ФГБОУ ВПО Национальный исследовательский университет

“Московский Энергетический Институт ”

Курсовая работа

По дисциплине “Программная инженерия”

“Классификация новостей с помощью методов

машинного обучения”

Выполнил:

Студент группы А-05-15

Миронов И.А.

Научный руководитель:

к. т. н. : Моросин О.Л.

Проверил:

к. т. н. доц.: Маран М.М.

Содержание

**Введение3**

**Основные определения и постановка задачи4**

**Анализ задачи6**

**Используемые библиотеки и инструменты7**

**Решение задачи8**

**Заключение22**

**Список литературы22**

Введение

Задача машинного обучения заключается в извлечении знаний из сырых данных и нахождения закономерностей между ними. Эта научная область находится на пересечении статистики, теории вероятности и компьютерных наук. Чаще всего алгоритмы машинного обучения используются для автоматизации процессов принятия решений путем обобщения известных примеров.

Одна из важнейших особенностей систем машинного обучения заключается в автоматизации поиска оптимального решения без вмешательства человека. Благодаря этому, человек смог изучать области, которые раньше были не известны в силу ограниченности человеческих возможностей. Были разработаны алгоритмы, способные определять эмоции человека, просчитывать прогноз погоды, диагностировать заболевания на ранних стадиях. С помощью них человек смог обрабатывать большие массивы информации за значительно меньшее время. С каждым годом алгоритмы становятся все совершеннее, количество информации больше, а входные данные все сложнее. Уже сейчас компьютер может понимать человеческую речь, распознавать объекты, прогнозировать исход событий до их наступления.

В данной курсовой работе будут рассмотрены методы, известные как обучение с учителем или контролируемое обучение (*supervised learning*).Их особенность заключается в том, что пользователь предоставляет алгоритму пары объект-ответ, а алгоритм способен выдать ответ для объекта, которого он никогда не видел раньше, без какой-либо помощи человека.

§2 Основные определения и постановка задачи

2.1 Определения

Объект – прецедент, по которому собраны некоторые данные[1].

Признак – набор характеристик, описывающий объект[1].

Обучение с учителем – алгоритм машинного обучения, при котором пользователь предоставляет алгоритму пары объект-ответ, а алгоритм находит способ получения ответа по объекту. [1]

Классификация – упорядоченное по некоторому принципу множество объектов, которые имеют схожие признаки, выбранные для определения сходства или различия. [1]

Основные виды задач классификации для обучения с учителем:

1. Бинарная классификация – классификация, в которой зависимая переменная может принимать только два значения. [1]
2. Многоклассовая (multi-class) классификация – классификация, в которой зависимая переменная может принимать любое конечное число значений[1].
3. Частичное обучение – задача, в которой для одной части объектов обучающей выборки известны и признаки, и ответы, а для другой только признаки[2].

Метрики качества:

1. Доля правильных ответов (accuracy) – считается, как сумма правильных ответов делить на общее количество[3].
2. F-мера:

TP (True Positives)— верно классифицированные положительные

TN (True Negatives) — верно классифицированные отрицательные объекты

FN (False Negatives) — положительные объекты, классифицированные как отрицательные.

FP (False Positives) — отрицательные объекты, классифицированные как положительные

Точность (precision) – ,

Полнота (recall) –

F-мера –

Получается, что F-мера есть гармоническое среднее точности и полноты[3].

2.2 Постановка задачи

Последнее время одной из основных проблем новостных агрегаторов стало распространение ложных новостей с очень высокой скоростью. Многие задаются вопросом, как можно обнаружить и блокировать данные новости на ранней стадии, чтобы предотвратить риск заблуждения граждан.

В данной курсовой работе ставится задача: разработать прототип алгоритма, позволяющего классифицировать новости с предварительной проверкой каждой на достоверность.

Полученный метод может значительно упростить и повысить качество работы новостных систем.

§3 Анализ задачи

Поставленная выше проблема относится к задачи мультиклассовой классификации. Это следует из того, что нужно разработать метод, который бы на основе входных данных, а именно: время публикации, рейтинг издания, заголовок статьи, страна происхождения и прочих признаков, смог бы проверить новость и отнести ее к одному из уже имеющихся классов.

Модель данной задачи:

Объект - новость

Класс - рубрики данного информационного портала

Цель - отладить метод обнаружения ложных новостей, выявить границы применимости

Примеры признаков:

* Номинальные: автор, издание, страна и тд.
* Порядковые: уровень доверия
* Количественные: скорость распространения, частота просмотров, упоминания в иных изданиях и тд.

Особенности задачи:

* Сверхбольшие выборки
* Разнотипные признаки
* Статья может относиться к нескольким рубрикам
* Не ясно, какие признаки вычислять по “сырым” данным

§4 Используемые библиотеки и инструменты

Для решения задачи будем использовать высокоуровневый язык программирования - Python 2.7. Его преимущество заключается в огромном количестве библиотек для анализа данных и возможности напрямую работать с программным кодом с помощью терминала или других инструментов типа Jupyter Notebook, который будет рассмотрен ниже

*Среда разработки:*

Jupyter Notebook - представляет собой интерактивную среду для запуска кода в браузере. Его особенностью является создание единого пространства, объединяющее код, текст и изображения[2].

*Используемые библио*т*еки:*

Scikit-learn – проект с открытым исходным кодом, содержащий ряд современных алгоритмов машинного обучения, а также полную документацию по ним[2].

NumPy - содержит функциональные возможности для работы с многомерными массивами и высокоуровневыми математическими функциями [2].

SciPy - набор функций для научных вычислений. Содержит процедуры линейной алгебры, математическую оптимизацию функций, обработку сигналов и статистические функции [2].

Matplotlib - основная библиотека для построения научных графиков[2].

Pandas - библиотека для обработки и анализа данных. Она построена на основе структуры данных, называемой DataFrame, которая представляет собой таблицу, похожую на электронную таблицу Excel[2].

Nltk (Natural Language Toolkit) - пакет библиотек для символьной и статистической обработки естественного языка[3].

§5 Решение задачи

5.1 Описание данных

Набор данных содержит текст и метаданные с 244 сайтов и представляет 12 999 постов за 2016 год. Данные были получены с помощью интернет-ресурса Kaggle, на котором и расположен датасет.[4]

*Описание признаков и целевой переменной:*

1. Uuid - уникальный идентификатор отправителя.
2. Ord\_in\_thread - количество потоков
3. Author - автор
4. Published - время публикации
5. Title - заголовок статьи
6. Text - текст статьи
7. Language - язык, на котором написана статья
8. Crawled - момент, когда статья стала популярна
9. Site\_url - сайт, разместивший новость
10. Country - страна регистрации информационного ресурса
11. Domain\_rank - уровень достоверности ресурсу
12. Thread\_title - Заголовок в других потоках
13. Spam\_score - процент ложных новостей в ресурсе
14. Main\_img\_url - ссылка на картинку новости
15. Replies\_count - количество ответов
16. Participants\_count - количество участников
17. Likes - количество пользователей, оценивших новость
18. Comments - количество комментариев
19. Shares - количество поделившихся записью
20. Type - ***целевая переменная:*** характеризует к какому классу относится новость

5.2 Первичный анализ данных

Для начала проверим наличие пустых значений в датасете.

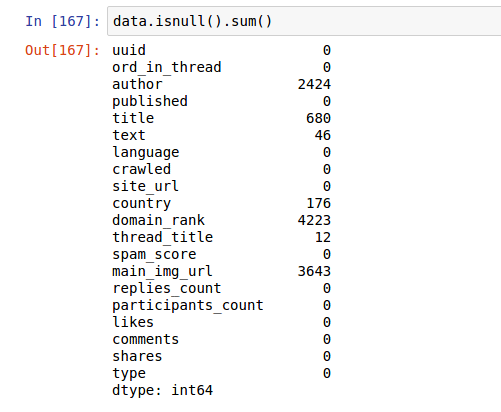


Рис.1 – Количество пустых значений по каждому признаку

Видно, что имеется большое количество незаполненных ячеек (см. рис.1). Для дальнейшего анализа это не допустимо, нужно их чем-то заменить.

Есть несколько подходов к решению этой проблемы:

1. Заменить пустые значения нулями
2. Заменить пустые значения средним по заданному столбцу
3. Заменить пустые значения на среднее K-ближайших соседей

В данной задаче нам важна любая информация о полученной новости и усреднять значение будет нелогичным, это повлечет потерю уникальности каждой новости и как следствие ухудшит обучающую модель. Поэтому заполним все отсутствующее элементы нулями (см. рис.2).

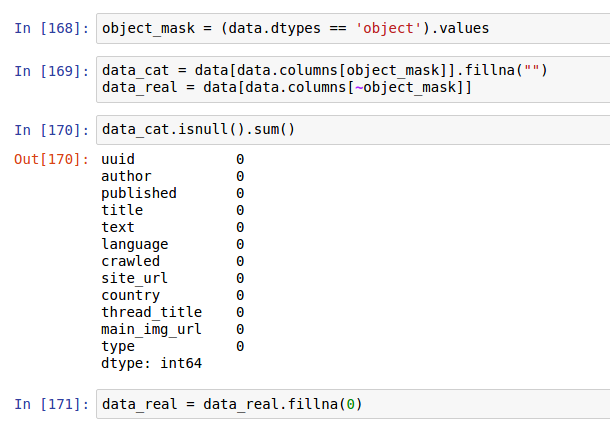


Рис.2 – Заполнение пустых значений

Также датасет был разделен на категориальные и числовые признаки. Это нужно для корректной обработки каждого типа признаков, так как подходы их анализа отличаются.

Целевая переменная имеет категориальный формат, с помощью встроенной функции LabelEncoder сопоставим каждому классу определенное число и посмотрим распределение классов (см. рис.3)[5].

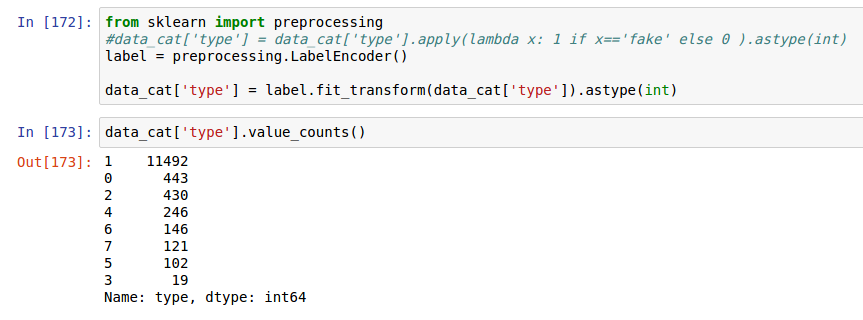


Рис.3 – Преобразование целевой переменой

Заметно, что значения целевой переменной сильно несбалансированны, более подробно это будет рассмотрено в параграфе 5.5.

5.3 Предобработка данных

Главная проблема в обработке данных заключается в том, что имеется большое количество текстовых переменных, которые нужно преобразовать в числовые, так как мы работаем с математическими функциями, на основе которых строятся обучающиеся модели.

Для начала рассмотрим заголовки новостей (в связи с ограниченностью вычислительных мощностей текст новости рассматриваться не будет).

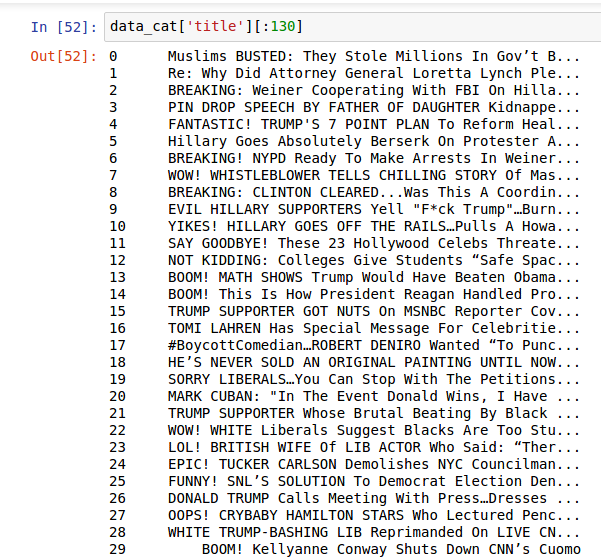


Рис.4 – Заголовки новостей

Как уже упоминалось выше, конечная цель - сопоставить каждому слову вектор. Однако видно, что текст слишком разнообразный(см. рис.4), поэтому чтобы минимизировать объем выходного датасета, проделаем несколько преобразований.

Первым делом очистим датасет от лишних символов. Нам интересны только слова, а вот, например, числа являются лишними, так как они несут количественную информацию, которая меняется на протяжение времени и никак не связана с классом новости. На рис.5 мы сначала очистили заголовки новостей от всех символов кроме букв и знаков препинания, а затем заменили все символы разделения предложений на точку для удобства (эмоциональный окрас в данном случае не важен).

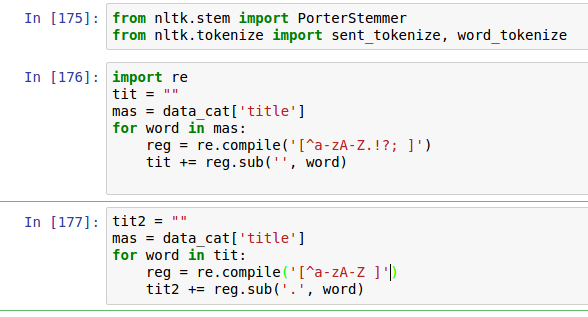


Рис. 5 – Очистка новостей

Для масштабирования данных рассмотрим подход, называемый TF-IDF(tern frequency-inverse document frequency).Идея этого метода заключается в том, чтобы присвоить больший вес термину, который часто встречается в конкретном документе, но при этом редко встречается в остальных документах корпуса. Если слово часто появляется в конкретном документе, но при этом редко встречается в остальных документах, оно, вероятно, будет описывать содержимое этого документа лучше. Само значение tf-idf для слова W в документе d вычисляется по формуле:

Где N – это количество документов в обучающем наборе,– это количество документов обучающего набора, в которых встретилось слово W, и TF – это частота встречаемости термина в запрашиваемом документе d [2].

Применим метод TF-IDF к заголовкам новостей (см. рис. 6).

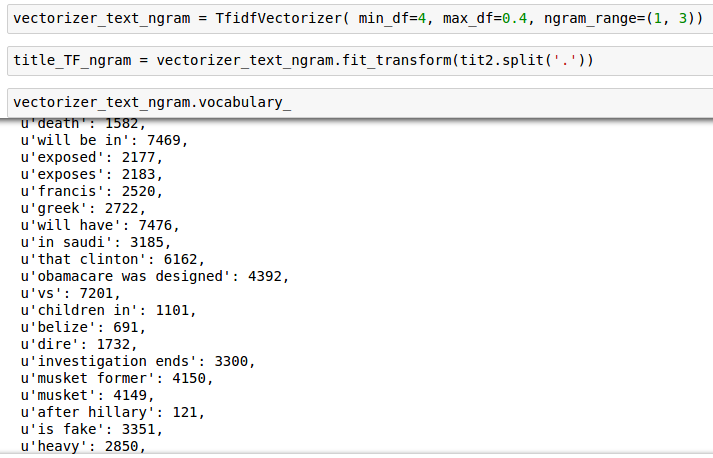


Рис.6 – Применение tf-idf по словам

В качестве параметров передадим Min-df - минимальное количество встреч слова, max-df - максимальное количество встреч (в процентах). Ngram\_range - передает из скольких слов высчитывать веса. В нашем случае считается частота встреч для словосочетаний от 1 до 3 слов.

Из полученных результатов можно сделать вывод, что данная модель имеет существенный недостаток. Слова в разных падежах и склонениях считаются как новый элемент и не суммируются с однокоренными словами, чего бы нам хотелось. Попробуем улучшить результат, используя для оценки не все слово, а только его часть. Для этого передадим в функцию TfidVectorizer параметр analyzer со значением char (см. рис. 7)

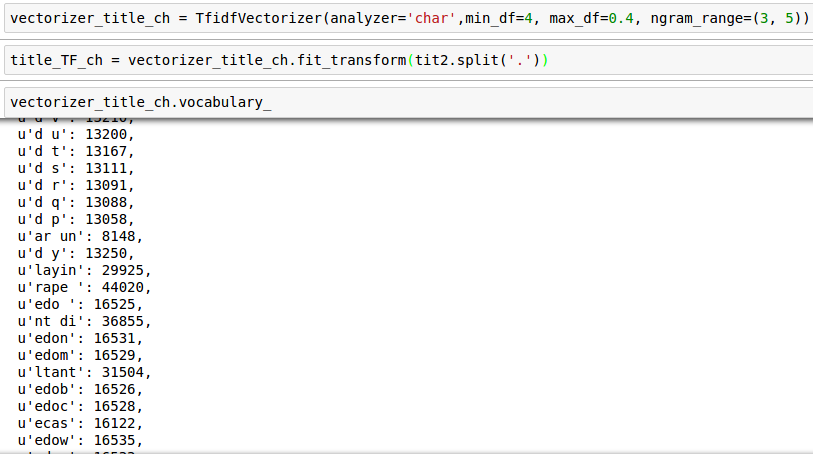


Рис. 7 –Применение tf-idf для символов

Видно, что теперь нам удалось анализировать не все слово, а лишь его часть, но из-за того, что мы не можем указать какую именно часть нужно взять для анализа, получаем не связный набор символов.

Следующий вариант решения этой проблемы заключается в предварительном выделении корней в предложениях и только потом применение метода tf-idf. Для этого воспользуемся встроенной функцией PorterStemmer (см. рис.8). Это алгоритм стемминга, основная идея которого заключается в последовательном применении ряда правил: отсечении окончаний и суффиксов, основываясь на особенностях языка[5].

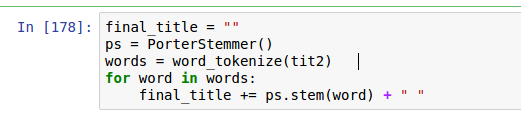


Рис.8 – Стемминг новстей

После стеммнига новостей снова применим метод TF-IDF(см. рис. 9)

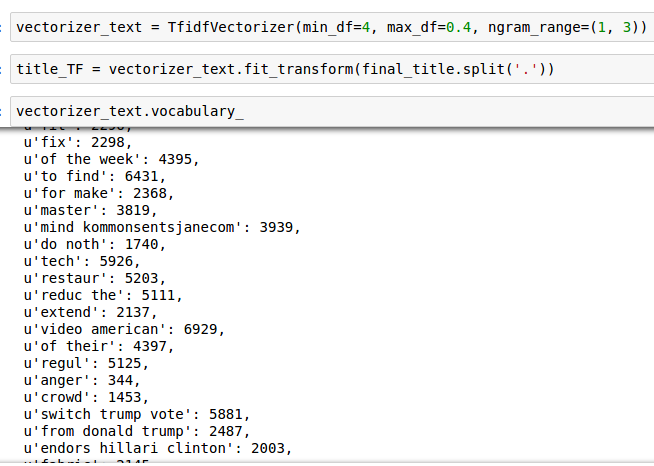


Рис.9 – Применение tf-idf после стемминга слов.

Теперь мы добились желаемого и в результате была получена матрица из коэффициентов.

Признак main\_img\_url переведем в бинарный, это поможет значительно сэкономить память и улучшит модель. Признаки uuid, thread\_title и text удалим, так как uuid уникален для каждой новости, thread\_title аналогичен title, а text мы не можем посчитать.

Теперь будут рассмотрены оставшиеся категориальные признаки(см. рис. 10).

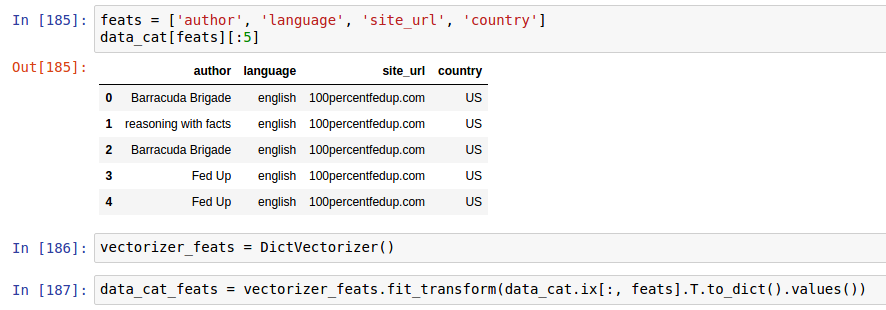


Рис.10 – Преобразование признаков

С помощью функции DictVectorizer преобразуем их в вектора[6]. Идея данной функции заключается в One-hot кодирование признаков, то есть замене категориальной переменной одним или несколькими новыми признаками, которые могут принимать значения 0 или 1.

Следующая проблема заключается в наличие временных признаков. Сначала преобразуем их в специальный тип Datetime. Затем вычтем из времени публикации время становления новости популярной и в результате получим промежуток “взлета” статьи(см.рис. 11).



Рис.11 – Преобразование временных признаков

Теперь можно удалить признаки crawled и published и добавить полученные числовые признаки в выходной датасет.

5.4 Визуальный анализ данных

На рисунке 13 представлена матрица корреляций числовых признаков. Корреляция показывает взаимосвязь двух или более объектов между собой. По умолчанию используется корреляция Пирсона, с расчетной формулой на рисунке 12[3]:

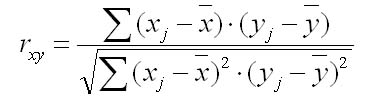


Рис. 12 – Формула корреляции Пирсона

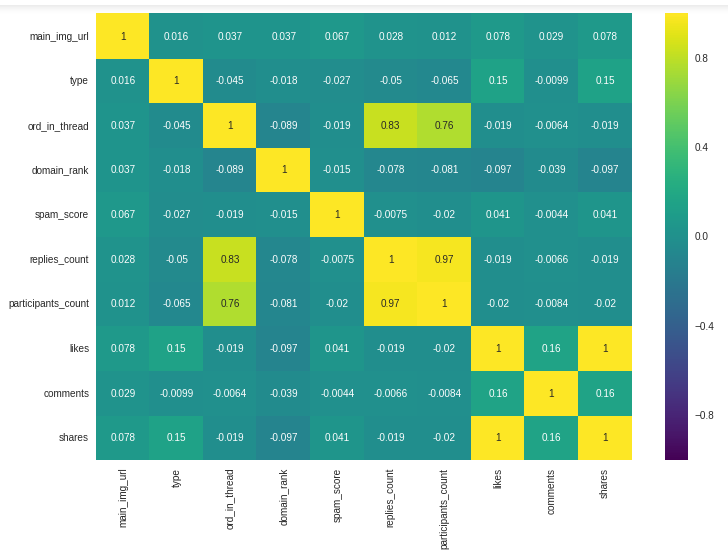


Рис.13 – Матрица корреляций

На рисунке 13 видна сильная корреляция shares и likes. Для более подробного анализа построим диаграмму от этих признаков (см. рис.14).

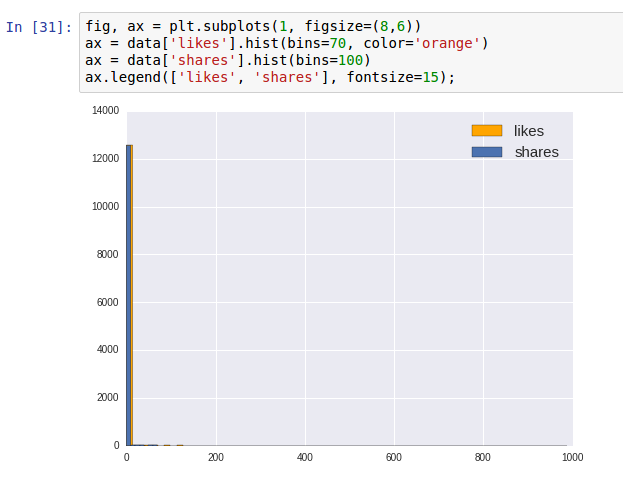


Рис.14 – Зависимость likes и shares

Как видно, эти данные сильно коррелируют из-за отсутствия значений в обоих случаях. С replies\_count и participants\_count ситуация аналогичная.

Уменьшим количество нулевых признаков объединив их и рассмотрим как распределяются новости по странам (см. рис. 15)

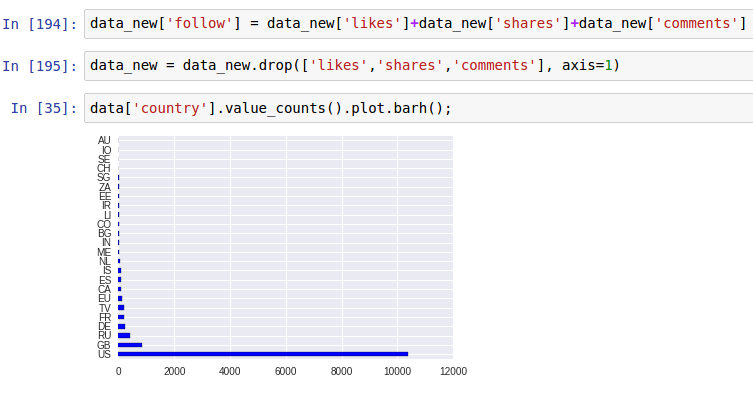


Рис.15 – Распределение новостей по странам

5.5 Обучение модели

При разбиение выборки будем использовать кросс-валидацию. Идея метода заключается в разбиение выборки на K блоков, затем модель обучается на K-1 блоке и происходит тестирование на оставшейся части данных. Процедура повторяется K раз, в результате получается оценка эффективности выбранной модели с наиболее равномерным использованием данных.

В связи с большим количеством итераций над категориальными признаками мы получили высокую дисперсию числовых данных. Это может сильно ухудшить обучающие модели, особенно линейные, поэтому произведем масштабирование признаков. Для этого будем использовать встроенную функцию StandartScaler (см. рис.16), хоть она и не делает распределение нормальным в строгом смысле слова, но в какой-то мере защищает от выбросов[6].

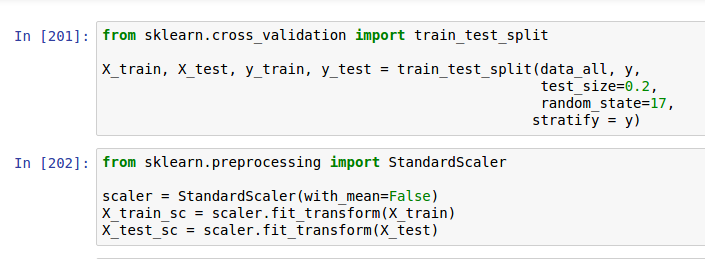


Рис.16

В качестве отправной точки построим дефолтовые модели логистической регрессии и случайного леса (см. рис.17)

Логистическая регрессия – это статистическая модель, используемая для предсказания вероятности возникновения некоторого события путём подгонки данных к логистической кривой, которая находится по формуле

Случайный лес – это множество решающих деревьев. В задаче регрессии их ответы усредняются, в задаче классификации принимается решение голосованием по большинству, Все деревья строятся независим[5].

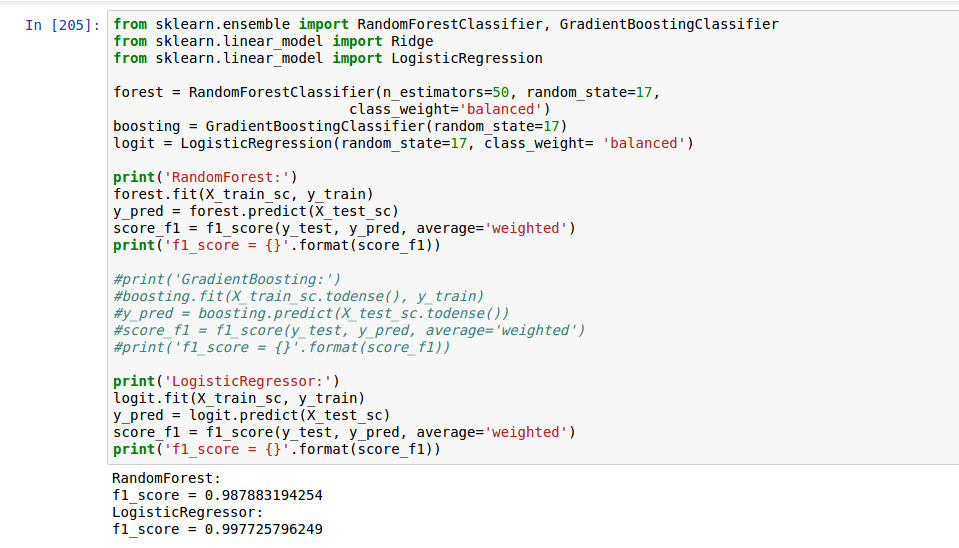


Рис. 17 – Построение обучающих моделей

Проверим, действительно ли получилось построить такую точную модель или она просто переобучилась. Для этого устроим перекрестную проверку по 5 блокам и оценим результат по каждому из них(см. рис. 18).

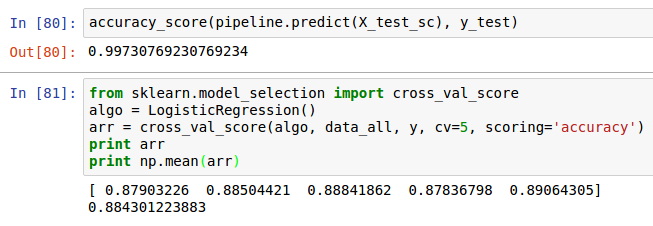


Рис. 18 – Перекрестная проверка

Сильное расхождение в результатах говорит о том, что модели переобучаются из-за не сбалансированности классов. Для борьбы с этим оценим важность каждого признака и удалим лишние из них (см. рис.19).

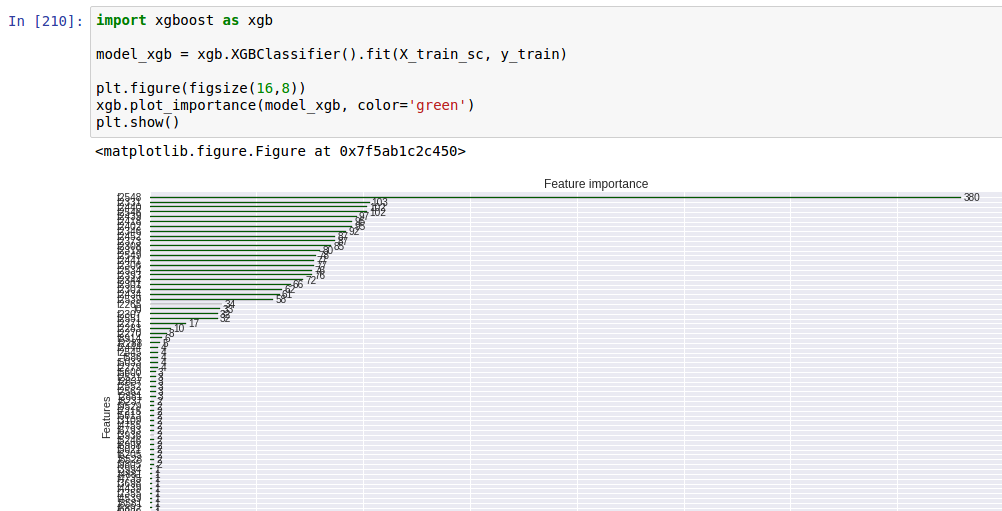


Рис.19 – Оценка важности признаков

Как видно на графике, есть примерно 50 главных признаков, а остальные являются посредственными - удалим их.

Для уменьшая размерности информации применим методом PCA(principal component analysis). Идея заключается в вычисление собственных векторов и собственных значений ковариационной матрицы исходных данных. Это поможет снизить корреляцию признаков между собой[2].

Воспользуемся моделью сверхслучайных деревьев (ExtraTreesClassifier), она более случайно вычисляет значения в узлах[6]. Алгоритм схож с методом случайных деревьев, но вместо поиска оптимальных порогов, значения выбираются произвольно для каждого признака, и наилучший из случайно генерируемых признаков выбирается, как правило, для разделения узла (см. рис. 20)

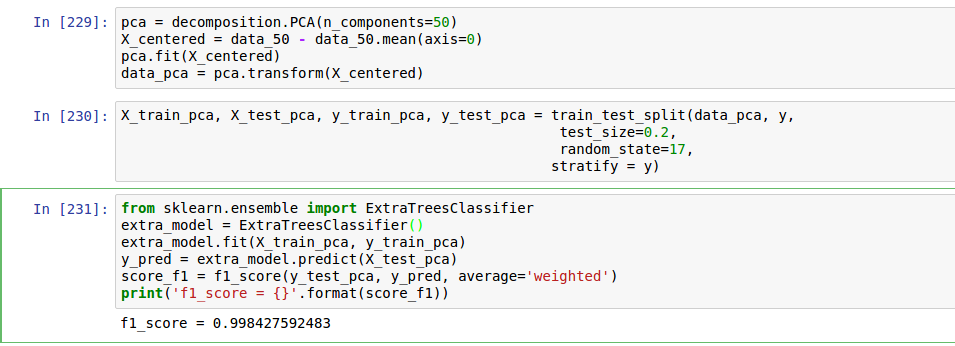


Рис.20 – Построение сверхслучайных деревьев

Теперь снова оценим точность полученной модели с помощью перекрестной проверки (см. рис. 21).

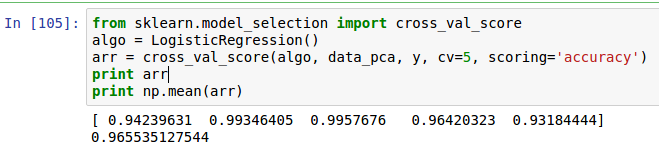


Рис. 21 – Перекрестная проверка после PCA

Видно, что теперь значения практически совпадают с результатами построенной модели, следовательно удалось снизить переобучаемость и повысить обобщаю способность алгоритма.

В основном, полученный результат был достигнут благодаря применению PCA. Хотя этот метод и показал очень хороший результат, но вместо него также можно попробовать применить факторизацию неотрицательных матриц (NMF) или множественное обучение с помощью алгоритмов t-SNE. Также можно значительно улучшить алгоритм стемминга подбирая наиболее оптимальные параметры для данного набора.

§6 Заключение

В данной курсовой работе был разработан прототип алгоритма для классификации новостей. Удалось преобразовать все данные к единому виду и обучить на них модель, с достаточно высокой точностью и минимальным переобучением. Обученная модель не является эталонной, но ее можно взять за основу для поиска более сложных решений данной задачи. Также стоит уделить внимание настройке гиперпараметров моделей, для повышения точности и посмотреть, как изменятся результаты при использование таких библиотек, как Vowpal Wabbit и Word2Vec.

К сожалению, проблема переобучения все еще остается открытой, для построения более объективной модели стоит сильно уменьшить размерность информации и увеличить дисперсию каждого класса, чтобы сформировать закономерности распределения, а не отталкиваться от исключений.

§7 Список литературы

[1] Грас Дж. Data Science. Наука о данных с нуля. – СПб.: БХВ-Петербург,2017.— 336с.

[2] Мюллер А., Гвидо С., Введение в машинное обучение с помощью Python. – М.: Вильямс, 2017 –480 с.

[3] Mohri M., Rostamizahed A., Talwalkar A., Foundations of Machine Learning — Cambridge MA: MIT Press, 2012. – 427 p.

[4]Kaggle competition [Электронный ресурс] // Интернет платформа для соревнований по анализу данных – 2017 – URL: <https://www.kaggle.com/mrisdal/fake-news> (дата обращения 24.11.2017)

[5] Коэльо Л., Ричарт В. Построение систем машинного обучения на языке Python – М.: ДМК Пресс, 2016. – 302 с.

[6]Scikit-learn documentation [Электронный ресурс] //официальная документация библиотеки Scikit-learn – 2017 – URL: <http://scikit-learn.org/> (дата обращения 24.11.2017).