Построение индекса сентиментов для анализа рынка акций

A Preprint

Бучко Даниил Владимирович
Факультет Экономических Наук
Студент, 3-ий курс бакалавриата
НИУ ВШЭ, г. Москва
dvbuchko@edu.hse.ru

Соколова Татьяна Владимировна
Факультет Экономических Наук
Старший преподаватель
Базовая кафедра инфраструктуры финансовых рынков
Аналитик в ЛАФР, НИУ ВШЭ, г. Москва
tv.sokolova@hse.ru

5 июня 2020 г.

Abstract

Когда заинтересованный исследователь попадает на финансовый рынок, для него открывается множество инструментов для принятия инвестиционных решений. Все эти инструменты в основе своей базируются на двух основновополагающих источниках: на фундаментальной и технической информации. Использование данных из этих источников подразумевает не только первичную обработку, но и последующую интерпертацию полученных результатов, что является достаточно трудозатратным процессом. А что если есть иные источники информации, которые могут передавать содержательный смысл анализа общедоступных данных? Что если направлять усилия не на исследования состояния определенной компании, а попробовать прислушаться к тому, что говорят люди об этой компании? Может быть у нас получится получить общее представление о состоянии компании, основываясь на разрозненные мнения? В данном исследовании я проведу анализ альтернативных источников принятия решений и проверю их эффективность на примере российского рынка акций. В качестве таких источников будут выступать форумы и телеграм каналы.

Вступление

Я бы хотел провести читателя этой статьи по пути вопросов и ответов, которые возникают сами собой в ходе решения любой поставленной задачи. Поэтому, сформулируем первый логичный вопрос: «Что мы хотим сделать?». Как было сказано раннее в асбстракте, мы бы хотели получить полезную информацию о компаниях, торгующих ценными бумагами, не прибегая к трудозатратным операциям, предполагающим знания финансовой отчетности или технического анализа. Следующий логичный вопрос: «Что это за информация и почему её можно считать ценной?» Частично на этот вопрос мы ответим в главе 1, где планируется выбрать данные и определиться с их источниками, а частично − в главе 4, где определимся с тем, насколько ценной оказалась полученная информация. В главе 1 я также рассмотрю особенности и способы получения данных. Будет показано на примерах, как выглядит структура источников, каким образом осуществляется автоматический сбор и обработка данных. Затем полученные данные мы начнем активно исследовать в главе 2, чтобы ответить на следующий вопрос: «Как использовать полученные данные?» При подробном расммотрении я опишу основные тонкости и механизмы предобработки полученных текстов, покажу особенности конкретно наших данных, посчитаю статистики и визуализирую основные результаты. В главе 3 мы займемся вопросами непосредственного моделирования языка при помощи машинного обучения и нейронных сетей: сформулируем задачу, выберем метрики, а затем подберем подходящую модель.

1 Получение данных.

Какие данные использовать?

Информацию об окружающем мире можно черпать из различных источников. Очевидно, что любой источник информационного потока имеет свои особенности и отличительные черты. К примеру, информация, публикуемая в журналах, обычно проходит через долгие корректировки, и к моменту выпуска издания, она может частично или полностью искажаться, утрачивать свою значимость и актуальность, в зависимости от задачи, в которой эта информация планировалась использоваться. Аналогичными свойствами можно охарактеризовать информацию, получаемую из новостей, регулярных аналитических сводок и стриминговых сервисов. Более того, важной отличительной чертой всех этих источников является то, что информация, получаемая на выходе, проходит через множество рук и любые конкурентные приемущества связанные с ценностью новой информации могут быть утеряны, потому что всегда будет некий посредник между исследователем и событиями.

Поэтому, для нашей задачи,– оперативно получать потенциально полезную информацию о финансовом рынке – необходимо выбрать источники, обладающие подходящими свойствами:

1. Релевантность. Важнее всего, чтобы информация с нашего источника имела отношение к ценным бумагам и гипотетически имела предсказательную силу. Ведь именно для этого мы и собираемся её собирать и обрабатывать. Отсекаем любые неэкономические источники данных.
2. Доступность. Это свойство позволит оперативно получать факты об изменениях на рынке без дополнительных ограничений, накладываемых посредниками или высокими издержками. Исключаем источники данных, за которые необходимо платить.
3. Высокая частотность данных. Очень удобно выбирать такой источник, информация с которого была бы в достаточном количестве, для проведения простейшего статистистического анализа. Кроме этого, это свойство дает возможность моментально адаптироваться к любым шокам, меняющим конъюктуру рынка ценных бумаг. Концентрируемся на постоянно обновляющихся, высокочастотных источниках данных.

Учитывая все аспекты, приведенные выше, я решил остановиться на текстовых источниках данных. Релевантность кокретизируется в текстовых данных, описывающих ситуацию на рынке акций. Как быть с доступностью? Здесь можно выбрать источники, не требующие денежных и временных издержек на ожидание получения информации – финансовые форумы и группы в социальных сетях. Заметим, что такие источники удовлетворяют и последнему свойству – свойству высокой частотности. Многие онлайн-платформы, ставшие привычным местом для обсуждения инвестиционных идей и стратегий существуют на российском рынке аж с конца 2000-х годов и активно развиваются по сей день.

1.1 Интернет сообщества: сайты

Безусловным лидером на на российском рынке финансовой информации является интернет платформа MFD. Судя по информации об интернет-ресурсе[1], эта платформа работает с 1996 года и объединяет ежедневно более 10000 человек. Вторым кандидатом на подходящий источник информации является интернет сообщество Smart-Lab. Помимо указанных ресурсов, существует несколько интересных организаций, предоставляющих финансовую информацию, среди них: Cbonds, Quote РБК, InvestFunds. Единственное отличие последних состоит в том, что они работают по принципу предоставления контента, а значит не до конца удовлетворяют нашей предпосылке о доступности. Остановимся на MFD.

1.2 Интернет сообщества: телеграм каналы

Телеграм-каналы стали популярны на пике введения блокировок в результате отказа основателя компании от предоставления ключей шифрования государству[2]. Сложно судить, что стало ключевым фактором в успехе телеграма, будь то зажигающая идея противостояния государству, которую так любит молодежь, либо удобный интерфейс, но телеграм полюбили и не обошли стороной люди, имеющие непосредственное отношение к финансовым рынкам. Так или иначе, телеграм предоставляет удобный и простой способ обсуждения различных тематик, поэтому заприметим этот источник информации

[1] https://mfd.ru/about/
[2] https://www.sostav.ru/publication/telegram-38688.html

как задел на будущие исследования и вернемся к его обсуждению в заключительной секции этого исследования.

Итак, мы разобрались с тем, что интернет площадка MFD обладает всеми необходимыми для нашего исследования характеристиками. Следующий вопрос: «Какую конкретно информацию, мы собираемся использовать?». Существуют многочисленные исследования в области анализа влияния текстовой информации на изменение в доходности ценных бумаг[3]. В данных исследованиях проверялась гипотеза о влиянии социальных сетей на доходность акций и индексов. Более того, были предприняты успешные попытки построить регрессионные модели, объясняющие доходность индексов, ориентируясь только на сентименты сообщений в социальных сетях. Попробуем сделать подобное исследование, но с использованием более современного подхода в области моделирования языка и на российском рынке акций.

1.3 Сбор информации с интернет-ресурсов

Поскольку мы определились с тем, что и откуда будем собирать, настало время понять, каким образом выполнить эту задачу. Для этого нужно получить представление о строении сайтов и понять, из каких деталей состоят веб-страницы. Ниже можно видеть переписку участников финансового форума в той форме, в которой её видит человек и в том обличии, в каком она предстает перед браузером.

Рис. 1: Каким образом информация предоставлена на сайте

Заметим, что каждое сообщение находится в соответствующем блоке (контейнере). Таким блокам присваивается метка «message», явно указывающая на содержимое контейнера. Весь процесс сбора данных состоит в том, чтобы отыскать все контейнеры на странице, получить их содержимое и перейти к следующей странице. Обратим внимание на то, как хорошо структурирована информация на этом примере. В реальности сайты редко бывают статическими, обычно они меняют свою структуру под воздействием пользовательский действий. Но это явно не случай российских форумов, написанных с конца 90-х годов. Кроме того, нужно с осторожностью пользоваться автоматическим сбором данных. Некоторые сайты запрещают использование парсеров, потому что слишком частые запросы страниц нагружают сервера и неправильно написанный код может значительно затруднить пропускную способность интернет-ресурса. Переходим к следующему шагу.

[3]Работы [1], [2], [3] в списке литературы

Таблица 1: Примеры полученных сообщений

Дата сообщения	Имя пользователя	Текст сообщения
09.11.2014 15:46	Веном	ШОРТИТЕ!!!
09.11.2014 17:17	михаил2	ты первый
09.11.2014 17:43	петрович	Так вы будете бить рекорд погружения или как?
09.11.2014 18:09	Веном	Рубль непоколебим в России. Доллар растёт
09.11.2014 18:27	capitan	Начинайте мне еще пару тысяч прикупить надо...
10.11.2014 14:33	capitan	Прикупил себе PLZL.
10.11.2014 15:47	capitan	Главное не слейся раньше времени как на Алросе) хотя я и сам так сделал)
10.11.2014 16:37	драконорожденный	Не не сольюсь. А с Алросой я ошибку сделал. Надо было смотреть на то
11.11.2014 11:36	драконорожденный	Думаю что эту цену мы увидим не раньше следующей весны...Я предупреждал
14.11.2014 22:52	МарКс	Все после таких новостей понятно стало
15.11.2014 22:59	трейдерсрублевки	http://oilru.com/news/436599/

2 Подготовка данных

В результате сбора данных были получены сообщения с 2008 по 2020 год по компаниям из разных групп капитализаций: маленькие, средние и большие. (см. примеры Таблица 1). Перед тем, как перейти к процессу построения моделей, необходимо предобработать текст таким образом, чтобы максимально исключить всю лишнюю информацию. Для этого текст, при помощи регулярных выражений, очищается от любой пунктуации, любых небуквенных символов, лишних пробелов и неинформативных ссылок и слов. После этого, каждое предложение приводится к нижнему регистру, лемматизируется (то есть приводится к своей начальной форме) и стеммингуется (обрезается таким образом, чтобы от слова осталась только его основа). Произведенные операции позволяют сократить количество неинформативных признаков, тем самым улучшая скорость и точность работы оптимизационных алгоритмов. В результате подобной обработки, сообщения стали иметь следующий вид:

Таблица 2: Сообщения после обработки

Дата сообщения	Имя пользователя	Текст сообщения
09.11.2014 15:46	Веном	шорт
09.11.2014 17:17	михаил2	ты перв
09.11.2014 17:43	петрович	так вы быт бит рекорд погружен ил как
09.11.2014 18:09	Веном	рубл непоколебим в росс доллар раст
09.11.2014 18:27	capitan	начина я ещ пар тысяч прикупа
10.11.2014 14:33	capitan	прикупа себ plzl
10.11.2014 15:47	capitan	главн не слива ран врем как на алрос хот я и сам так сдела
10.11.2014 16:37	драконорожденный	не не слива а с алрос я ошибк сдела быт смотрет на то
11.11.2014 11:36	драконорожденный	дума что этот цен мы увидет не ран след весн я предупрежда
14.11.2014 22:52	МарКс	посл так новост понятн станов
15.11.2014 22:59	трейдерсрублевки	

Обратим внимание на то, что сообщения в большинстве случаев не потеряли свой основной смысл, однако количество уникальных слов сократилось в 4 раза: с $131,530$ до $34,733$. Такой подход в обаботке текстов имеет свои ограничения, и иногда останавливаются на этапе лемматизации, но как мы увидим далее, именно такой способ подготовки сообщений позволит делать наиболее точные прогнозы в рамках конкретно нашей задачи. В приложении можно посмотреть на другие интересные статистики по датасету (см. Главу 5).

Как можно видеть из рисунка 2, в результате обработки данных частотность слов распредилилась более равномерно, уменьшая долю выбросов в выборке. На изображение не попали слова с очень низкой

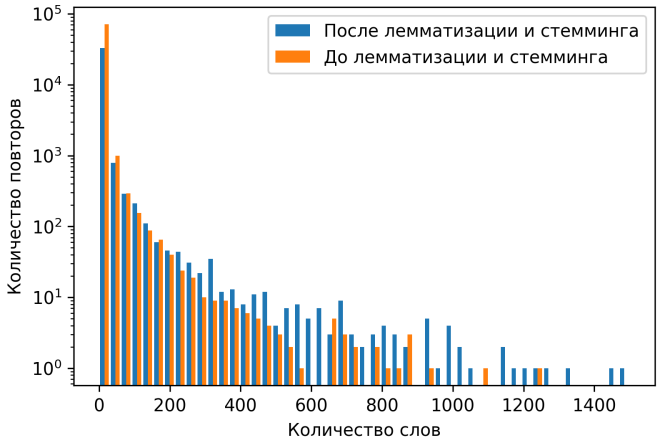

Рис. 2: Распределение слов в тренировочном наборе данных

частотностью, для сохранения наглядности. В приложении есть таблица 3, показывающая примеры частоупотребляемых и редкоупотребляемых слов. Переходим к выбору модели.

3 Выбор модели

Следующий момент, который мы должны для себя решить – как использовать собранные данные? В работах [1], [2] авторы предпринимали попытки объяснить доходность бумаг, исходя из гипотезы, которую можно сформулировать следующим образом: «Эмоциональный фон, складывающийся вокруг определенной бумаги, влияет на решение инвестора (по крайней мере частного) о покупке или продаже той или иной бумаги». Для того чтобы проверить эту гипотезу, нам необходимо построить модель $F(.)$, которая бы переводила сообщения во множество эмоциональных признаков по следующей схеме:

$$F(\text{сообщение}) = \begin{cases} 3, & \text{если сообщение носит позитивный окрас} \\ 2, & \text{если сообщение не относится к финансовому рынку} \\ 1, & \text{если сообщение носит негативный эмоциональный окрас} \end{cases} \tag{1}$$

Получается, что для нас, как для исследователей, задача сводится к построению модели, разделяющей все сообщения на 3 группы. С точки зрения машинного обучения, такая задача именуется задачей классификации. Выделим ключевые особенности нашего набора данных, чтобы выбрать подходящую модель:

1. Изменчивость лексики во времени. Некоторые фразы, под влиянием различных культурных особенностей, меняют со временем то, как мы выражаем свои мысли, поэтому из всего количества собранных данных, для обучения были взяты сообщения из различных временных промежутков.
2. Постобработанные данные не имеют меток класса. Простым языком, чтобы использовать алгоритмы обучения с учителем, нам необходимо дополнительно вручную разметить данные. Особенности рутинной ручной разметки ограничивают количество данных для обучения, поэтому из этого пункта возникает следующая характерная черта.
3. Относительно небольшой датасет. В результате разметки совместными усилиями с группой единомышленников был сформирован итоговый датасет, состоящий из $30,000$ наблюдений.
4. Несбалансированность классов. Очень важная черта, которую обязательно необходимо иметь в виду, при построении любой модели. В наших данных после разметки, распределение классов имело вид, изображенный на рисунке 3

Особенности 3-4, обсуждаемые выше, будут активно приниматься во внимание при построении моделей.

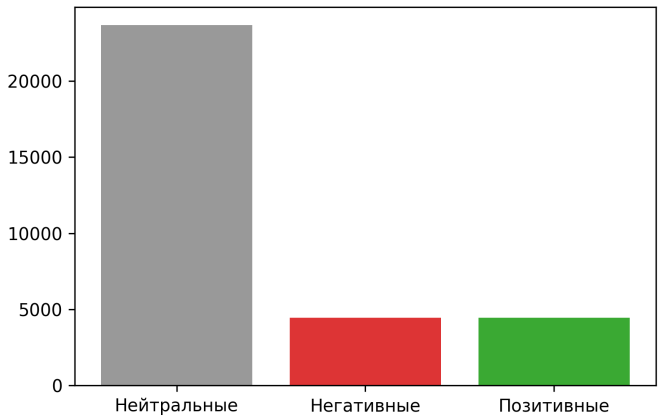

Рис. 3: Распределение категорий в тренировочном наборе данных

3.1 Представление данных

Чтобы построить классификатор сообщений, нужно представить данные, понятным компьютеру образом. Более формально, нам необходимо определить пространство, из которого искомая модель $F(.)$ будет переводить сообщения в соответсвующие сентиментам классы. Имея численное представление каждого из предложений, можно применять математический аппарат к построению моделей. Принимая во внимание методы машинного обучения и статистики, такие представления могут задаваться самыми различными способами. В ходе исследования мы будем постепенно двигаться от простейших к более продвинутым, попутно делая выводы о преимуществах и недостатках каждого из подходов.

3.1.1 OneHotEncoding

OneHotEncoding – один из множества способов определения множества значений, на котором будет работать искомая модель $F()$. Суть метода заключается в создании дамми-переменных в количестве, равном количеству слов в датасете. Таким образом, каждое сообщение можно закодировать вектором, состоящим из нулей и единиц. Единицы будут характеризовать наличие определенного слова, а нули – отсутствие. Такой подход накладывает свои ограничения, среди которых:

1. Необходимость хранения в оперативной памяти большого объема данных в процессе обучения алгоритма. В нашем случае, размер датасета, закодированного таким образом, имел бы размерность
$$R^{\text{Количество сообщений} \times \text{Количество слов}} = R^{31000 \times 34733}$$
2. Учитывая, что каждое слово кодируется единицей, получаем, что все слова имеют одинаковый вес. Однако, в русском языке некоторые слова встречаются гораздо чаще других, и их большее количество в тренировочных данных будет накладывать отпечаток на процесс оптимизации, о котором речь пойдет ниже. В качестве шага предобработки были предприняты меры по уменьшению количества лишних слов, но как можно видеть в таблице 3, среди наиболее употребительных слов все еще мелькают слова, не влияющие на сентимент сообщения.

3.1.2 Tf-Idf Encoding

Tf-Idf – решение проблемы равновзвешивания слов в процессе моделирования связи между сообщениями и их эмоциональным окрасом. Согласно этому подходу, каждому уникальному слову в тренировочном наборе данных соответствует определенный вес, исчисляемый по специальной формуле. Взвешивание слов должно решать главную проблему – уменьшать значимость слов, которые не несут практического смысла. Как определить какое слово несет смысл, а какое нет? Создательница[4] TF-IDF предложила

[4]Karen Spärck Jones. Synonymy and Semantic Classification — Edinburgh University Press., 1986. — Vol. 1. — (Edinburgh Information Technology series)

следующую идею: если слово характерно не только для определенного сообщения, а для всего датасета в целом, то скорее всего, оно несет меньшую смысловую нагрузку, чем слово, которое встречается только в одном сообщении датасета. Чтобы отделить такие слова друг от друга, она придумала статистику, под названием TF-IDF, которая состоит из двух частей: TF (Term Frequency) и IDF (Inverse Document Frequency) и считается для каждого слова в отдельности.

$$\text{TF-IDF}_\text{i} = TF_\text{i} \times IDF_\text{i} \quad \forall i \text{ из словаря всех слов}$$

TF_i часть отвечает за важность слова в пределах одного сообщения и рассчитывается по следующей формуле:

$$TF_i = \frac{n_i}{\sum\limits_{k=1}^{n} n_k}$$

где n_i− количество повторений i-го слова в сообщение, а n_k−количество слов в одном сообщении.
IDF_i отвечает за значимость слова в контексте всех сообщений в том смысле, о котором говорилось ранее:

$$IDF_i = \log \frac{|D|}{|\{d_t \in D | i \in d_t\}|}$$

где $|D|$−количество всех сообщений, а $|\{d_t \in D | i \in d_t\}|$−количество сообщений, в которых встретилось i-ое слово. Получается, что для каждого слова мы считаем величину IDF, которая является единой для одинаковых слов и TF, которая разнится от предложения к предложению. Величина $TF - IDF$ тем больше, чем характернее слово для каждого из предложений. Среди проблемных мест такого численного представления слов можно выделить следующий:

1. При кодировании теряется значение слов, определяемое порядком. К примеру, для TF-IDF, равно, как и для OneHotEncoding не существует разницы между предложениями «Здесь сложно придумать что-то совершенно не бессмысленное» и «Здесь не сложно придумать что-то совершенно бессмысленное».

Разобравшись с возможностями представления слов в численном виде, мы смогли наглядно увидеть, как можно подготовить текстовые данные к тому, чтобы начать оценивать заветную модель (1). Настала пора перейти к самой интересной[5] части нашего исследования − подбору модели.
Одно из негласных правил хорошего тона при любых исследованиях в сфере машинного обучения − начинать с простейших моделей. Эта идея понятна: чем сложнее задача, чем сильнее желание попробовать что-то из ряда вон выходящее, но при рассмотрении сложных моделей, требующих более тонкой надстройки, исследователи, занимающиеся процессом оптимизации, зачастую не понимают, как хорошо работает их алгоритм. Можно видеть значения точности в районе 70% успешно классифицируемых сообщений, однако совершенно непонятно, является ли это значение наилучшим, или, может быть, необходимо тратить кучу дополнительного времени на сбор дополнительных сообщений и их последующей разметки[6]. Начнем выбор модели с основных понятий.

3.1.3 Основные понятия

Подбор модели осуществляется в три этапа:

1. Выбор оптимизируемой функции потерь. Эта специальное название функции, которая нужна, чтобы понимать, как хорошо наша модель отражает моделируемую зависимость. Функция потерь позволяет обучать алгоритмы.

[5]Когда я проходил стажировку в одной из компаний занимающихся Data-Science, мой руководитель любил говорить, что выбор модели - награда за долгую и рутинную работу по предобработке данных.

[6]Andrew Ng - известный исследователь в области машинного обучения и нейронных сетей. Его видеокурсы доступны на платформе coursera.org. На одной из своих лекций он рассказывал о компаниях, занимающихся применением ИИ в своих разработках, которые обращались к профессору за консультациями по исследованиям. Разработчики рассказывали о задаче, об алгоритмах которые они применяли, о результатах оптимизации и находились серьезные компании, которые не могли понять, почему их модели выходят на плато по точности и огромные временные и денежные затраты по добыче новых данных не исправляют проблем. Решением подобных ситуаций оказывалось сравнительная конкурентоспособность продвинутых алгоритмов с простейшими моделями. Руководители ИИ компаний, используя простейшие алгоритмы, получали бэнчмарки, нижние пороги по качеству моделей и понимали, что правильное направление - дооптимизация существующих алгоритмов.

2. Выбор метрики качества. Эта метрика необходима, чтобы сравнивать между собой различные алгоритмы и интерпретировать потенциал обученной модели.
3. Выбор непосредственно модели. Модель − попытка формального построения связи между объектами исследования.

3.1.4 Функция потерь

В парадигме подхода машинного обучения, подбор модели, отражающей исследуемую зависимость, осуществляется на основе постоянных наблюдений за влиянием изменений параметров модели на качество предсказания. Простыми словами, мы каким-то образом перебираем набор моделей и пытаемся понять, как хорошо та или иная модель отражают действительность. Учитывая особенности конкретно нашей модели (1), нам нужно учитывать, что сама модель переводит сообщения из некоторого пространства произвольной рамерности в 3-х мерное пространство эмоциональных состояний. В идеале, нам нужно предсказывать вероятности каждого из классов, причем предсказания каждого из классов образуют полную группу событий. Поэтому, избираемая функция потерь должна учитывать эту особенность. Кроме этого, вспомним об особенностях наших данных, заключающихся в несбалансированности классов. На такие запросы теория машинного обучения предлагает следующую функцию:

$$Loss(y, \hat{p}) = -\sum_{c=1}^{3} w_c y_c log(\hat{p}_c), \quad \text{где}$$

$w \in R^3-$ вектор весов.
$y \in R^3-$ бинарный вектор метки класса определенного сообщения
$\hat{p} \in R^3-$ вектор предсказания вероятностей сентимента для определенного сообщения

Напомню, что размерность вектора ответов для предсказываемого сообщения равна 3, потому что мы предсказываем вероятности каждого из 3-х классов сентимента. Вектор весов будет учитывать ошибку, совершаемую на несбалансированном классе сильнее, чем на доминирующем, чтобы мы не делали поспешных выводов о моделях с низкой функцией потерь, постоянно предсказывающих доминирующий класс. Итак, подытожим: процесс обучения будет сводиться к оптимизации функции потерь, путем перебора различных моделей. Чем меньше вероятность верного класса, тем выше значение функции потерь.

3.1.5 Метрика качества

Хорошая метрика качества должна оценивать предсказательную силу модели и давать ей интерпретацию. В случае многоклассовой классификации, предлагается следующий набор метрик:

1. Accuracy - показатель средней точности распознавания класса. Его основной недостаток заключается в отсутсвии чувствительности к несбалансированности. Высчитывается по следующей формуле: $Accuracy(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}[y_i = \hat{y}_i]$
2. F-мера - статистика, не обладающая недостатком первой метрике. Высчитывается по формуле:

$$\text{F-мера} = 2\frac{precision \times recall}{precision + recall}$$

Остановимся поподробнее на F-мере. Представим, что после обучения на тренировочных данных мы хотим понять, как хорошо наша модель научилась видеть зависимости между словами и их сентиментами. Для этого мы делаем прогноз на отложенных данных и смотрим на результирующую матрицу ошибок, имеющую следующий вид:

Матрица ошибок	Негативный	Нейтральный	Позитивный
Негативный	96	3	0
Нейтральный	6	650	1
Позитивный	6	2	115

В матрице ошибок по столбцам описано количество реальных таргетов, а на их пересечении со строками − количество предсказаний отнесенными классификатором к соответсвующему строке классу. Например,

число 96 в матрице можно интерпетировать следующим образом: модель распознала 96 негативно окрашенных сообщений из 108. Построив такую матрицу, мы сможем определить две вещи: 1) точность алгоритма - доля сообщений, принадлежащих к верному классу; 2) полнота - доля предсказанных сообщений для верного класса в общей величине предсказанных классов. Первая вещь есть ничто иное, как Precision, а вторая - Recall. Контроль не только точности предсказания, но и отсутствия спутанности по предсказанию других классов объединяется в гармоническом среднем двух оценок - в F-мере. Такая форма объединения показателей гарантирует, что F-мера будет маленькой, если хотя бы один из показателей (Precision или Recall) будет маленький. Очень удобно.

3.1.6 Выбор моделей

Регрессия, как бэнчмарк

Итак, обсудим, какими качествами должна обладать подходящая для нашей задачи модель:

1. Переводить численное представление слов в вектор размерностью R^3, характеризующий вероятность каждого из классов. Поскольку сообщения должны быть классифицированы однозначно, сумма вероятностей каждого из классов складываются в единицу .
2. Иметь набор параметров, изменяя которые, можно влиять на предсказания модели.

Как и было заявлено в начале исследования - начинаем с простейших моделей. Рассмотрим основной регрессионный подход: Пусть мы имеем матрицу данных размерностью $R^{31000\times34733}$. Напомню, что 31000−количество сообщений, а 34733− количество уникальных слов в этих сообщениях. Для каждого из 31000 сообщений мы имеем вектор-строку вероятностей каждого из класса, например для нейтрального класса вектор-строка вероятностей имеет вид:

$$p = (1 \quad 0 \quad 0)$$

В действительности, если верен первый класс, то истинная вероятность того, что сообщение также принадлежит позитивному классу равна нолю. Это важный момент, потому что данная идея наводит на мысль о том, что какие бы предсказания в трехмерном пространстве не строила бы наша модель, значения соответсвующих вероятностей должны быть отнормированы таким образом, чтобы при суммировании вероятностей для каждого из наблюдений мы получали строгую единицу. Для этой цели в машинном обучении была придумана функция под названием «Softmax»

$$Softmax(z_i) = \frac{e^{z_i}}{\sum\limits_{i=1}^{3} e^{z_i}}, \quad \text{где}$$

$z \in R^3-$ вектор предсказаний, построенный нашей моделью

Простейшая модель, генерирующая такой трехмерный вектор предсказаний является нейронная сеть, как на рисунке ниже. Предложения подаются в числовом представлении, одним из способов (OneHotEncoding или TF-IDF). Затем эти вектора умножаются на матрицы весов, таким образом, чтобы в результате умножения мы получили трехмерный вектор. Затем значения, выдаваемые этим трехмерным вектором нормируются softmax функцией. Весь этот цикл операций называется прямым проходом (forward pass) и математически представляется в следующем виде:

$$\hat{p}_i = Softmax(X_i W + \beta) \tag{2}$$

$$\hat{p}_i \in R^{(1,\ 3)}, \quad X_i \in R^{(1,\ 34733)}, \quad W \in R^{(34733,3)}, \quad \beta \in R^{(1,3)} \tag{3}$$

$$Loss(y_i, \hat{p}_i) = -\sum_{c=1}^{3} y_{c_i} log(\hat{p}_{c_i}) \tag{4}$$

Отметим, что $\hat{p}_i$ - вектор вероятностей для i-го сообщения. После прохождения прямого прохода, мы считаем ошибку при помощи формулы (4) и обновляем параметры W и β таким образом, чтобы уменьшать значение функции ошибки на следующей итерации обучения. Способ обновления весов носит название градиентного спуска и предполагает обновление весов в соответствии с градиентом функции ошибки относительно каждого из параметров W и β.

Процесс обновления весов формально:

$$W_{t+1} = W_t - \alpha \frac{\partial Loss(W_t, \beta_t)}{\partial W_t}$$

$$\beta_{t+1} = \beta_t - \alpha \frac{\partial Loss(W_t, \beta_t)}{\partial \beta_t}$$

Такой подход обеспечивает постепенное уменьшение оптимизируемой функции потерь до определенного момента. Двигаясь в сторону антиградиента, рано или поздно мы дойдем до локального минимума функции ошибок. Параллельно с обучением, в конце каждой итерации происходит контроль значения функции ошибки на отложенной выборке, чтобы наблюдать за обощающей способностью алгоритма.

Дерево решений

Еще одни алгоритм, который мы будем использовать помимо регрессионного подхода - использование дерева решений. Без лишних подробностей скажу только, что дерево работает по следующему принципу: на каждом итерационном шаге, алгоритм перебирает все слова из поданного на вход предложения и выбирает те, которые уменьшают кросс-энтропия путем полного перебора всех возможных слов. Кроме обычного дерева решений, существует случайный лес, состоящий, как можно было догадаться, из множества деревьев решений. У случайного леса есть одно важное преимущество - модель не запоминает данные, а вырабатывает обобщающую способность с ростом сложности оцениваемой структуры.
Сказав достаточно об используемых моделях перейдем к основным показателям, полученным в результате исследования:

	One Hot Encoding				TF-IDF			
	Acc_{train}	Acc_{test}	$F1_{train}$	$F1_{test}$	Acc_{train}	Acc_{test}	$F1_{train}$	$F1_{test}$
Нейросеть	0.853	0.724	0.771	0.559	0.804	0.732	0.690	0.567
Лес	0.984	0.736	0.977	0.530	0.989	0.744	0.983	0.507
Дерево	0.785	0.554	0.741	0.458	0.890	0.614	0.856	0.472

Как можно видеть из результатов, наилучшие показатели по нашей основной метрике качества F-мере демонстрирует нейросеть. Можно было бы выбрать её в качестве основной модели и использовать, но здесь есть один очень важный момент. Что общего между всеми этими подходами? И нейронная сеть, и дерево решений, и случайный лес,- все модели используют в качестве данных статистики, полученные из тренировочного датасета. Простым языком, если мы будем использовать слова, которые отличаются от слов, присутствовавших в исходном датасете (не формой слова, а написанием), то будем терять информацию с этих слов, просто потому что ни One Hot Encoding пространство, ни TF-IDF не способны дать существенную информацию о новых словах. Как же нам быть? На помощь приходят современные нейросетевые алгоритмы. В 2013 году миру была представлена модель[7], изменившая представление вещей на долгие годы вперед. Группа исследователей из компании Google выяснили, что если, имея все знания мира, попытаться отдать их нейросети в текстовой форме, то последняя сможет построить векторное пространство всех слов в соответствии с их семантическим взаимотношением. Конечно, понятия «отдать знания нейросети» подразумевают постановку специальной задачи и куда более специфичные технические тонкости, но основной смысл состоит в том, что в 2013 году мир получил возможность черпать знания о словах извне. Есть прекрасные современные ресурсы, демонстрирующие то, что я имею в виду[8]. Как это открытие повлияет на наше решение? Очень просто: отныне и впредь мы будем переводить слова в векторы не при помощи One Hot Encoding или TF-IDF подхода, а при помощи предобученных векторов. Мы будем использовать эти предобученные вектора с той целью, чтобы при появления синонимичного слова, отсутствовавшего в нашем тренировочном датасете, модель не выкидывала что-то нелогичное, а относилась бы к этому слову, как к синониму какого-нибудь другого слова из тренировочного набора данных. Надо сказать, что само по себе добавление знания о внешнем мире не увеличило точность модели на отложенной выборке, но при подробном приближении, мы заметили, что необычные слова классифицируются правильнее с использованием обученного векторного пространства, чем при помощи OHE или TF-IDF. Последние имели тенденцию к «занейтраливанию» сообщения.

[7]Distributed Representations of Words and Phrases and their Compositionality, https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[8]https://rusvectores.org/ru/

4 Тестирование полученной информации

Создав языковую модель перевода сообщений в их сентименты, мы вольны что-угодно творить с полученными рядами данных (см. таблицу 4 в приложении). Для начала выдвенем основные гипотезы, относительно возможных связей полученных данных со стоимостью котировок. Логично предположить, что если люди говорят много хорошего про бумагу, то скорее всего она будет расти. Аналогичное верно и для обратного: если сказано много плохого, то скорее всего стоимость бумаги начнет падать. Кроме того, вероятно, степень влияния сентиментов сильнее для компаний с маленькой капитализацией, по сравнению с компаниями большей капитализации. Данные гипотезы мы будем тестировать при помощи симуляции портфелей.[9] В ходе этого исследования проводилась симуляция портфелей ценных бумаг, отобранных по специальным метрикам, которые, в свою очередь, базировались на показателях сентиментов. Симуляция портфеля проходила в следующие этапы:

1. Отбирались 60 компаний. Отбор происходил по размеру компаний: по 20 штук для малой, средней и высокой капитализации.
2. На основе сентиментов строились фильтры, согласно которым, каждый месяц определялись лидеры в каждой из групп капитализации.
3. Компании-лидеры покупались и держались ровно месяц. За этот месяц фиксировался результат доходности портфеля, после чего следовала ребалансировка (повторение шага 2).

Какие же фильтры использовались для ребалансировки портфелей?

Спецификация фильтров

4.0.1 Количество положительных сообщений

4.0.2 Дивергенция мнений

4.0.3 Количество отрицательных сообщений

4.0.4 Доля сообщений по компании в величине всех сообщений

4.0.5 Среднеквадратичное отклонение доходности внутри месяца

4.0.6 Суммарный объем торгов по акции

4.0.7 Изменение цены закрытия к предыдущему периоду

[9]Идея тестировать гипотезы таким образом принадлежит Александру Томтосову, моему коллеге по научной лабаратории. Моя роль в симуляции заключалась в автоматизации процессов.

5 Приложение

Таблица 3: Частотность первых и последних 20 слов после обработки

Слово	Частотность слова	Слово	Частотность слова
не	13517	marketbeat	1
а	6964	bmo	1
эт	4206	canad	1
год	2402	raymond	1
шорт	1807	james	1
быт	1654	canaccord	1
сегодн	1626	genuit	1
ден	1626	cibc	1
так	1517	mull	1
нефт	1507	торонт	1
рост	1499	акр	1
цен	1442	мобилизова	1
акц	1334	скептичн	1
дума	1265	финанасов	1
пок	1241	недопустим	1
див	1219	гсп	1
рынок	1184	xi	1
дава	1163	неаффилирова	1
сво	1159	неаффилирован	1
компан	1059	несоответств	1

Рис. 4: Простейшая нейронная сеть

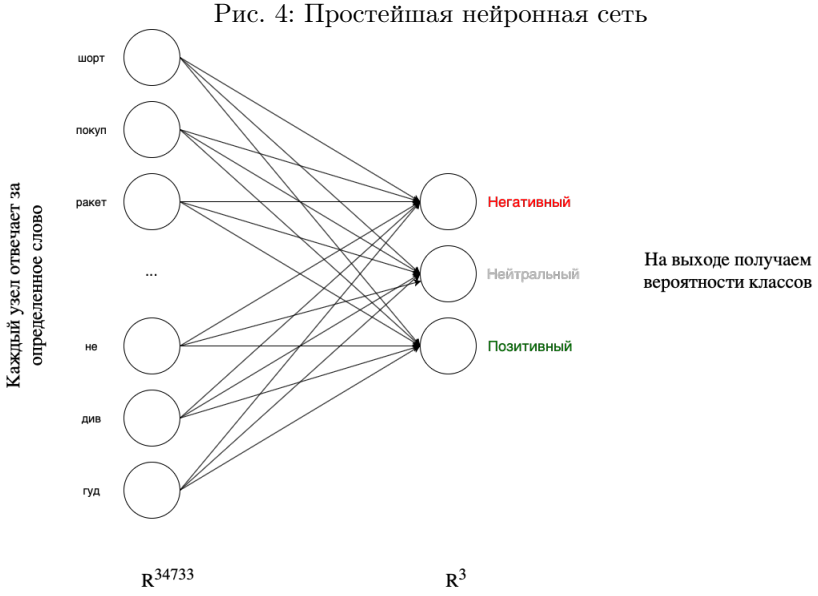

Таблица 4: Данные после распознавания сентимента

Дата	Негативных	Нейтральных	Позитивных
2013-02-11	44	67	18
2013-02-12	154	263	89
2013-02-13	110	215	59
2013-02-14	112	147	47
2013-02-15	35	74	20
2013-02-16	18	34	11
2013-02-17	8	21	6
2013-02-18	44	121	30
2013-02-19	43	80	25
2013-02-20	17	50	12
2013-02-21	18	14	12
2013-02-22	57	55	13
2013-02-23	3	10	0
2013-02-24	0	2	0
2013-02-25	29	77	22
2013-02-26	27	73	12
2013-02-27	23	85	20
2013-02-28	24	66	6
2013-03-01	15	29	11
2013-03-02	7	14	6

Таблица 5: Группы компаний для анализа

Маленькая капитализация		Средняя капитализация		Крупная капитализация	
Компания	Объем	Компания	Объем	Компания	Объем
НКНХ	176.5	Уралкалий	248.0	Газпром	4766
Черкизово	75.7	Распадская	32.6	Сбербанк	4786
Ленэнерго	67.6	ЛСР	63.5	Лукойл	3778
Иркут	42.2	Мечел	41.4	Норникель	3156
Ятэк	33.0	Фосагро	364.7	Татнефть	1297
Аптеки 36 и 6	32.0	Ростелеком	234.5	Сургутнефтегаз	1686
Селигдар	21.8	Россети	320.5	Роснефть	4134
КТК	15.7	Русгидро	307.5	ВТБ	1001
Сар. НПЗ	14.5	Система	165.8	Алроса	490.1
Русолово	11.8	ТМК	61.4	Мосбиржа	268.3
Соллерс	8.81	Юнипро	175.2	Яндекс	945.1
ГАЗ	7.29	Акрон	241.3	Аэрофлот	95.5
Ашинский	2.01	ПИК	255.6	ФСК ЕЭС	234.1
ЧЗПСН	1.38	Лента	79.3	ММК	474.5
Дагсбыт	0.51	Мосэнерго	86.2	МТС	645.3
Арсагера	0.40	ТГК-1	50.3	Новатэк	3253
GTL	0.24	Русал	409.1	НЛМК	829.5
Тантал	0.24	Транснефть	1196	Северсталь	790.3
Роллман	0.12	Газпромнефть	1636	Магнит	390.4
Сиб. Гостинец	0.04	НМТП	182.0	Полюс	1420

Список литературы

[1] Nuno Oliveira, Paulo Cortez, Nelson Areal. The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices In Expert Systems With Applications 73 (2017), pages 125–144.

[2] Thomas Renault. Intraday online investor sentiment and return patterns in the U.S. stock market In Journal of Banking and Finance, 2017 84th, pages 25–40. .

[3] Thi-Thu Nguyen and Seokhoon Yoon. A Novel Approach to Short-Term Stock Price Movement Prediction using Transfer Learning In Journal Applied Sciences, 2019 9th, pages 3–16.