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

Московский институт электроники и математики им. А.Н.Тихонова

Направление подготовки/специальности
«01.03.04 Прикладная математика»
Образовательная программа **«Прикладная математика»**

О Т Ч Е Т
о прохождении
производственной практики

Студент _____Чиков А.П._____БПМ171_____ _____
(Фамилия И.О.) номер группы

Руководитель практики студента:

ООО "1С"	**Старичков Н.Ю.**	**разработчик**
должность и место работы	Фамилия И.О.	должность

Руководитель практики от НИУ ВШЭ:

	Внуков А.А.	*к.т.н.,PhD,доцент*
должность и место работы	Фамилия И.О.	должность

Практика пройдена с оценкой _____9_____

Дата ____31.07.2020_____

Москва, 2020

Оглавление

1 Введение

В рамках производственной практики были выделены следующие цели:

1. Знакомство с инструментами, технологиями и алгоритмами машинного обучения
2. Получение опыта использования их на практике

Для достижения указанных целей были поставлены следующие задачи:

1. Разработать систему классификации сообщений электронной почты по классам, указанным пользователем
2. Предусмотреть возможность пользовательской настройки алгоритма классификации
3. Протестировать полученную систему классификации

Разработка системы классификации велась на языке программирования Python с использованием библиотек для обработки естественного языка (NLTK) и машинного обучения(sklearn). Для предварительного анализа данных и оценке качества полученного классификатора использовался Ernon Email Dataset [1].

2 Кратка характеристика организции

Фирма "1С" основана в 1991 г. и специализируется на разработке, дистрибьюции, издании и поддержке компьютерных программ делового и домашнего назначения.

Из собственных разработок фирмы "1С" наиболее известны программы системы "1С:Предприятие", а также продукты для домашних компьютеров и образовательной сферы.

Система программ "1С:Предприятие" предназначена для автоматизации управления и учета на предприятиях различных отраслей, видов деятельности и типов финансирования, и включает в себя решения для комплексной автоматизации производственных, торговых и сервисных предприятий, продукты для управления финансами холдингов и отдельных предприятий, ведения бухгалтерского учета ("1С:Бухгалтерия" самая известная учетная программа в ряде стран), расчета зарплаты и управления кадрами, для учета в бюджетных учреждениях, разнообразные отраслевые и специализированные решения, разработанные самой фирмой "1С", ее партнерами и независимыми организациями.

Система "1С:Предприятие" широко распространена в России и странах СНГ, успешно применятся организациями многих стран мира. Постановлением Правительства России от 21 марта 2002 года за создание и внедрение в отраслях экономики системы программ "1С:Предприятие" коллективу разработчиков – сотрудников "1С" была присуждена Премия Правительства РФ в области науки и техники.

Фирма "1С" является официальным дистрибьютором деловых программных продуктов зарубежных и отечественных производителей, таких как Microsoft, Лаборатория Касперского, Eset, ABBYY, DrWeb, Аскон, ПроМТ, Entensys, Novosoft и другие.

Фирма "1С" также выступает издателем программных продуктов ведущих отечественных разработчиков на территории России. Проект "Издания 1С:Дистрибьюция" был запущен в 2004 году и включает в себя на сегодняшний день продукты таких производителей, как ABBYY, Лаборатория Касперского, Acronis, Aladdin, ASP Linux, ALT Linux, Entensys, Redline Software, Dragon Soft, Infotecs, Movavi, Paragon Software, Paragon Mobile, Famatech, SmartLine, Oxygen Software, VITO Technology, Panda Security, Infowatch. Список продуктов в линейке изданий "1С:Дистрибьюция" постоянно пополняется.

"1С" работает с пользователями через разветвленную партнерскую сеть, которая включает более 10 000 постоянных партнеров в 600 городах 25 стран [2]

3 Постановка задачи

В рамках производственной практики необходимо было разработать скрипт для классификации писем пользователя на заданные им категории. Процесс получение итогового программного продукта был разделен на три этапа:

1. Изучение Gmail API [3] и написание скрипта для взаимодействия классификатора и почтового ящика пользователя.
2. Изучение алгоритмов машинного обучения, методов предобработки и векторизации текста и дальнейшее написание классификатора для писем электронной почты с использованием изученных технологий.
3. Тестирование полученного программного продукта на предмет его работоспособности и качества итоговой классификации.

Для выполнения поставленных задач был выбран итеративный подход, представляющий собой трехэтапный цикл:

1. Изучение теоретической составляющей рассматриваемой технологии – просматривание материалов, статей, специализированных сайтов для получения теоретических знаний о применении выбранной технологии для дальнейшего ее использования.
2. Практическое исследование рассматриваемой технологии – для практического освоения выбранной технологии и оценки ее влияние на качество итоговой модели классификации технология испытывалась на тестовом датасете с письмами (Ernon Email Dataset).
3. Внедрение технологии в итоговую модель классификации

4 Полученные результаты

В данном пункте будут описаны результаты, полученные в ходе разработки проекта, в соответствии с постановкой задачи, указанной в пункте 3. Основное внимание будет уделено процессу разработки алгоритма классификации.

4.1 Gmail API

Для взаимодействия с почтовым ящиком пользователя, расположенном на платформе GMail, был разработан скрипт, реализующий следующий функционал с использованием GMail API:

1. Авторизация пользователя – для каждого пользователя создается отдельная папка, в которой будут храниться его идентификационные данные, письма и другая служебная информация
2. Выгрузка писем пользователя с почтового ящика в локальное хранилище
3. Извлечения текста писем и других параметров (отправитель, заголовок и т.д)
4. Выгрузка пользовательских ярлыков
5. Настройка ярлыков для автоматической классификации
6. Настройка параметров уверенности классификации
7. Загрузка ярлыков для автоматически классифицированных писем

Взаимодействие пользователя со скриптом происходит с использованием консольного интерфейса.

4.2 Классификация пользовательских писем

Алгоритм классификации включает в себя два основных этапа: предобработка писем и извлечение из них признаков для классификации и обучения модели на обработанных данных. В данном пункте будет кратко описан круг изученных технологий обработки текста и представлен разработанный алгоритм классификации.

4.2.1 Нормализация

На первом этапе работы с текстовыми данными их необходимо нормализовать – привести к однородному виду. Для нормализации текста использовались следующие методы:

1. Приведение слов к нижнему регистру
2. Удаление из текста не буквенных символов
3. Токенизация текста – разделение текста на отдельные слова и удаление знаков пунктуации.
4. Стемминг – усечение слов текста до их основы.

4.2.2 Векторизаця

Следующим этапом работы с текстовыми данными является их векторизация – извлечение признаков для дальнейшей классификации. Для векторизации нормализованных текстовых данных рассматривались следующие методы:

1. **BOW**(Bag-of-words) – представление текста в виде набора слов, которые в нем встречаются, с сохранением количества их вхождений
2. **BOW_binary** – вариация стандартного BOW, использующая вместо количества вхождений слова в текст лишь факт его вхождения. Для лучшего понимания рассмотрим пример:

   Текст1 = “Мама мыла раму. Даша мыла раковину”
   Текст2 = “Мыла раковину мама”

	мама	мыла	раму	даша	раковину
Текст1	1	2	1	1	1
Текст2	1	1	0	0	1

BOW

	мама	мыла	раму	даша	раковину
Текст1	1	1	1	1	1
Текст2	1	1	0	0	1

BOW_binary

3. **TF-IDF**(***TF*** *— term frequency,* ***IDF*** *— inverse document frequency*) – представление текста в виде вектора оценок важности каждого слова в контексте текста, являющегося частью коллекции текстов. Более формально (w – слово, t – текст, T – множество текстов, n_w – количество вхождений слова w в текст t):

$$TF(w,t) = \frac{n_w}{|t|}$$
$$IDF(w,T) = log(\frac{|T|}{|\{t \in T | w \in t\}|})$$
$$TF - IDF(w,t,T) = TF(w,t) \times IDF(w,T)$$

4.2.3 Выбор классификатора

Алгоритм классификации, который будет описан в следующем пункте, предполагает использование двух моделей классификации, применяемых последовательно для “популярных” и “непопулярных” классов. Модель классификации состоит из трех компонент:

1. **Векторизатор** – рассматривалось три варианта векторизации, описанные в пункте 4.2.2

2. **Классификатор** –полный список рассматриваемых моделей приведен в таблицах с результатами тестирования

3. **Стратегия классификации**:

   a) **OvO** (One-vs-One) – обучение одного бинарного классификатора для каждой пары классов. Предсказание класса для очередного объекта происходит на основе голосования: для каждого из классов считается количество классификаторов, которые его предсказали, и среди всех классов выбирается один с наибольшим количеством голосов.

   b) **OvR** (One-vs-Rest) – обучение одного бинарного классификатора на каждый из имеющихся классов. Для предсказания класса очередного объекта выбирается классификатор с наибольшим показателем уверенности классификации.

Каждая тройка “векторизатор - классификатор - стратегия” оценивалась средним значением метрики *accuracy* на кросвалидации. В Таблице 1 приведены результаты тестирования для одного из пользователей:

	Strategy	TF-IDF	BOW	BOW(binary)
KNeighborsClassifier	ovo	0,749	0,562	0,443
	ovr	0,770	0,586	0,522
LogisticRegression	ovo	0,512	0,706	0,682
	ovr	0,567	0,718	0,730
GaussianNB	ovo	0,689	0,677	0,617
	ovr	0,524	0,519	0,455
MultinomialNB	ovo	0,507	0,713	0,679
	ovr	0,524	0,699	0,722
ComplementNB	ovo	0,555	0,708	0,687
	ovr	0,581	0,696	0,722
RandomForestClassifier	ovo	0,579	0,617	0,629
	ovr	0,699	0,684	0,727
SVC(linear)	ovo	0,715	0,689	0,696
	ovr	0,761	0,706	0,725
SVC(rbf)	ovo	0,601	0,584	0,550
	ovr	0,701	0,624	0,634
SVC(sigmoid)	ovo	0,708	0,517	0,629
	ovr	0,756	0,452	0,737

Популярные категории

	Strategy	TF-IDF	BOW	BOW(binary)
KNeighborsClassifier	ovo	0,539	0,320	0,320
	ovr	0,633	0,461	0,367
LogisticRegression	ovo	0,500	0,437	0,460
	ovr	0,539	0,515	0,523
GaussianNB	ovo	0,618	0,523	0,491
	ovr	0,524	0,273	0,195
MultinomialNB	ovo	0,391	0,437	0,499
	ovr	0,516	0,531	0,585
ComplementNB	ovo	0,477	0,437	0,507
	ovr	0,539	0,523	0,585
RandomForestClassifier	ovo	0,352	0,351	0,406
	ovr	0,648	0,648	0,655
SVC(linear)	ovo	0,570	0,406	0,484
	ovr	0,672	0,531	0,601
SVC(rbf)	ovo	0,454	0,211	0,297
	ovr	0,625	0,492	0,578
SVC(sigmoid)	ovo	0,500	0,195	0,305
	ovr	0,688	0,085	0,679

Непопулярные категории

Таблица 1 Результаты тестирования для пользователя “*mcconnell-m*”.

Зеленым отмечены три лучших результата для каждого из методов веторизации. В ходе тестирования выяснилось, что лучшие тройки для различных пользователей могут различаться, поэтому для выбора оптимальной модели тестирование производилось для всех пользователей с усреднением полученных результатов. Результаты тестирования приведены в Таблице 2.

	Strategy	TF-IDF	BOW	BOW(binary)
KNeighborsClassifier	ovo	0,695	0,589	0,548
	ovr	0,723	0,610	0,581
LogisticRegression	ovo	0,613	0,721	0,726
	ovr	0,648	0,738	0,747
GaussianNB	ovo	0,717	0,719	0,698
	ovr	0,636	0,637	0,597
MultinomialNB	ovo	0,539	0,679	0,643
	ovr	0,624	0,739	0,728
ComplementNB	ovo	0,604	0,677	0,644
	ovr	0,690	0,735	0,729
RandomForestClassifier	ovo	0,605	0,626	0,634
	ovr	0,715	0,720	0,727
SVC(linear)	ovo	0,740	0,689	0,727
	ovr	0,772	0,706	0,737
SVC(rbf)	ovo	0,665	0,576	0,622
	ovr	0,721	0,666	0,681
SVC(sigmoid)	ovo	0,728	0,540	0,648
	ovr	0,766	0,505	0,718

Популярные категории

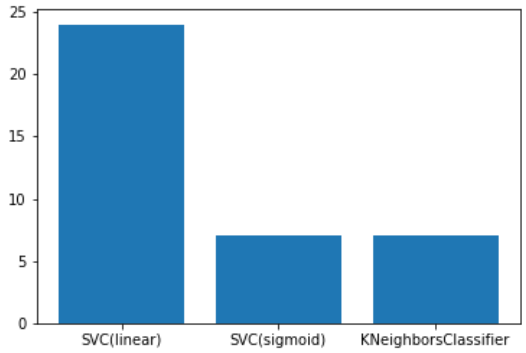
	Strategy	TF-IDF	BOW	BOW(binary)
KNeighborsClassifier	ovo	0,646	0,505	0,469
	ovr	0,718	0,526	0,493
LogisticRegression	ovo	0,620	0,592	0,606
	ovr	0,675	0,623	0,642
GaussianNB	ovo	0,733	0,670	0,628
	ovr	0,678	0,564	0,511
MultinomialNB	ovo	0,583	0,632	0,602
	ovr	0,690	0,668	0,628
ComplementNB	ovo	0,659	0,633	0,606
	ovr	0,724	0,663	0,629
RandomForestClassifier	ovo	0,541	0,537	0,559
	ovr	0,645	0,632	0,668
SVC(linear)	ovo	0,730	0,574	0,624
	ovr	0,755	0,621	0,655
SVC(rbf)	ovo	0,601	0,468	0,513
	ovr	0,696	0,587	0,600
SVC(sigmoid)	ovo	0,596	0,436	0,521
	ovr	0,721	0,460	0,671

Непопулярные категории

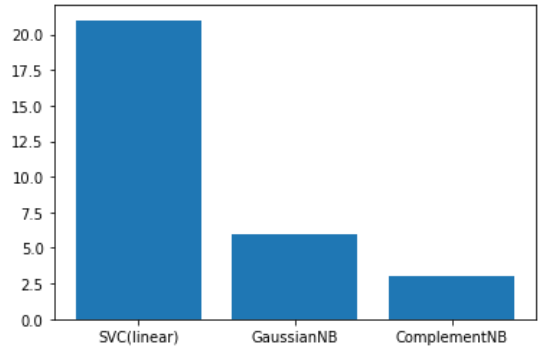
Таблица 2 Усредненные результаты тестирования для всех пользователей.

На основании результатов тестирования была выбрана следующая модель классификации:

1. **Векторизатор**: TF-IDF
2. **Стратегия**: OvR
3. **Классификатор**: для каждой и двух метакатегорий было выбрано три лучших модели. Для каждой модели в тройке было посчитано количество “выигрышей”, то есть количество раз, когда значение метрики *accuracy* на кросвалидации для данной модели было наибольшим в тройке. В качестве классификатора выбиралась модель с наибольшим показателем данной величины, в обоих случаях это оказалась SVC(linear). Результаты тестирования приведены на Графике 1.



Популярные категории

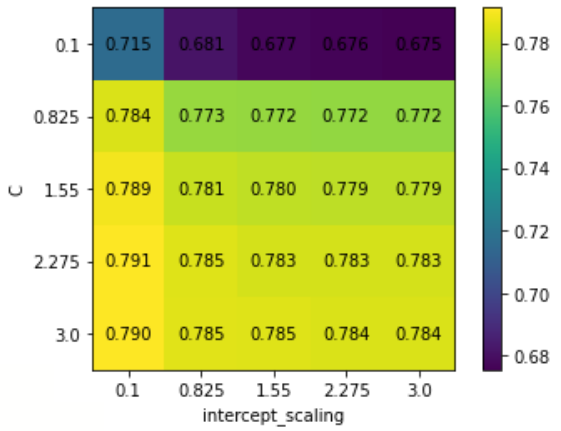


Непопулярные категории

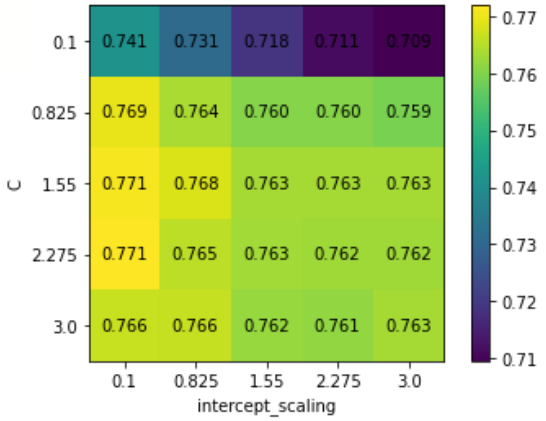
График 1 Количество “выигрышей” для различных моделей.

4.2.4 Настройка гиперпараметров

Для подбора оптимальных гиперпараметров модель SVC(linear) была заменена на ее аналог LinearSVC. Данная модель имеет больше возможностей для настройки параметров регуляризации. На основе результатов кросвалидации для нее были подобраны оптимальные гиперпараметры *C* и *intercept_scaling*, распределенные на регулярной решетке. Результаты подбора приведены на Рисунке 1.



Популярные категории



Непопулярные категории

Рисунок 1 Значение метрики *accuracy* для различных значений гиперпараметров.

Качество классификации настроенной модели оценивалось усреднением метрики *accuracy* на кросвалидаци для всех пользователей. Результаты тестирования приведены в Таблице 3.

	Популярные	**Непопулярные**
SVC(linear)	0,772	0,755
LinearSVC(default)	0,776	0,763
LinearSVC(tuned)	0,791	0,771

Таблица 3 Результаты тестирования настроенной модели.

4.2.5 Алгоритм классификации

В данном пункте будет приведено краткое описание алгоритма классификации писем пользователя и приведена мотивировка выбора данного алгоритма.
В результате исследования писем нескольких пользователей из Ernon Email Dataset были выявлены следующие особенности:

1. Количество писем у пользователя зачастую не превышает значения 500, то есть при выборе модели классификации и настройки ее гиперпараметров необходимо учитывать маленький размер обучающей выборки.

2. Категории писем пользователей можно условно разбить на две метакатегории: “популярные” – 10 писем в категории и более и “непопулярные” – от 3 до 10 писем в категории. Данные категории писем необходимо рассматривать раздельно.

3. Добиться качественной классификации писем из “непопулярных” категорий практически невозможно, поэтому для классификации подобных писем необходимо использовать дополнительные признаки, а также предусмотреть опцию отказа от автоматической классификации. Для этого в модель введены параметры *alpha* и *beta* – уверенность классификации, которые настраивает пользователь.

Таким образом, с учетом особенности данных был разработан следующий алгоритм классификации:

Обучение модели

1. Нормализация текстов писем
2. Обучающая выборка делится на две части, соответствующие метакатегориям “популярные” P и “непопулярные” NP.
3. Векторизация нормализованных текстов на основе TF-IDF для каждой из частей обучающей выборки.
4. Обучение модели классификатора, выбор которой был описан ранее:
    1) Для “популярных” категорий – обучение модели на веторизованных текстах писем.
    2) Для “непопулярных” категорий – обучения модели на веторизованных текстах писем с добавлением дополнительного набора признаков, все дополнительные признаки были подвергнуты минимакс стандартизации:
        a) Количество символов в тексте
        b) Длина самого длинного слова
        c) Средняя длина слов
        d) Количество знаков пунктуации
        e) Количество стоп-слов
        f) Отправитель

На Схеме 1 приведена схема описанного алгоритма.

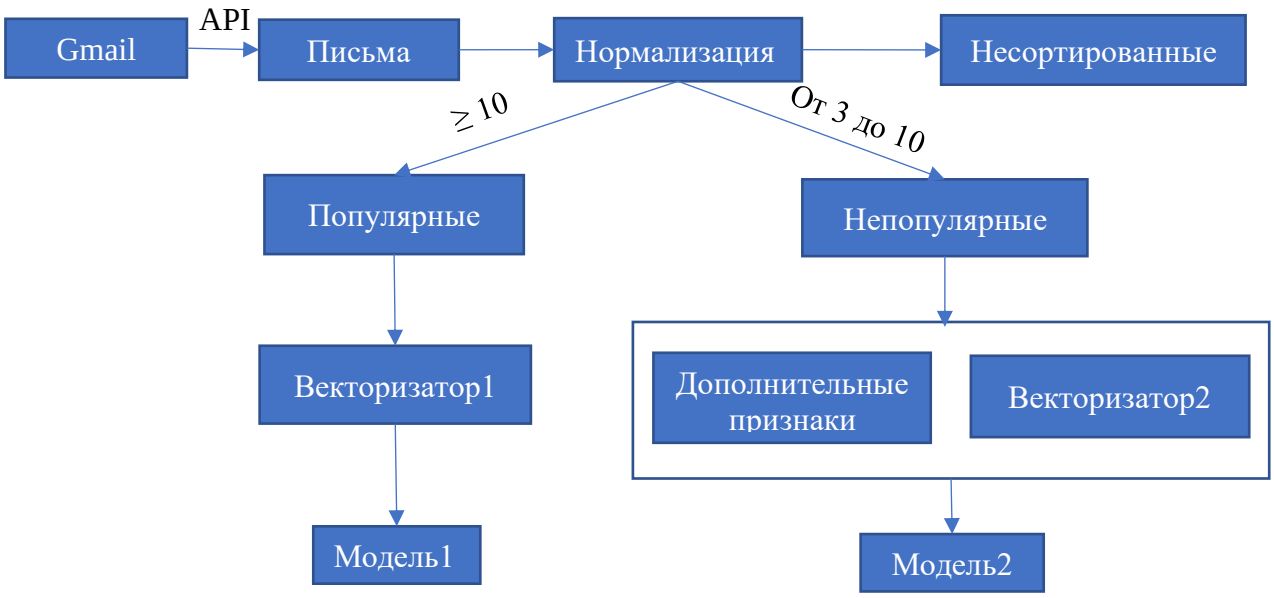


Схема 1 Алгоритм обучения модели.

Классификация

1. Нормализация текстов писем
2. Векторизация нормализованных текстов векторизатором, обученным на выборке P.
3. Модель, обученная на выборке P, предсказывает вероятность принадлежности писем к тому или иному классу
4. Для полученных вероятностей устанавливается порог alpha, таким образом если для письма вероятность принадлежности к определенному классу выше alpha, то письму присваивается этот класс, иначе письмо остается неклассифицированным.
5. Неклассифицированные письма векторизуются векторизатором, обученным на выборке NP. Для писем добавляются дополнительные признаки, описанные в пункте 4.2 в описании процесса обучения.
6. Модель, обученная на выборке NP, предсказывает вероятность принадлежности писем к тому или иному классу
7. Для полученных вероятностей устанавливается порог beta, таким образом если для письма вероятность принадлежности к определенному классу выше beta, то письму присваивается этот класс, иначе письмо остается неклассифицированным.
8. Новые метки классов подгружаются в пользовательский почтовый ящик

На Схеме 2 приведена схема описанного алгоритма.

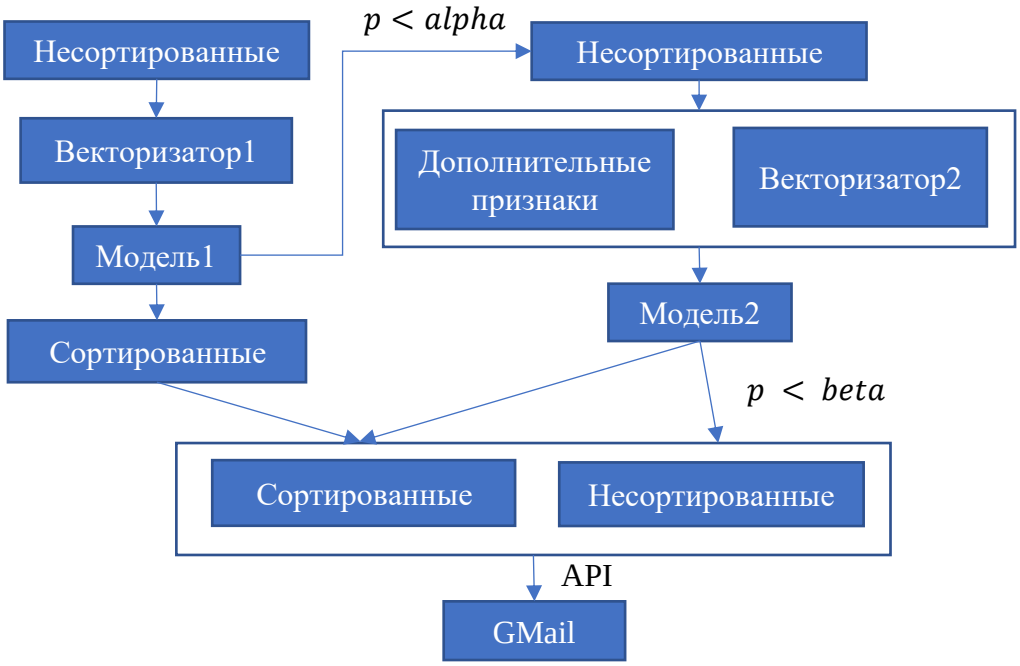
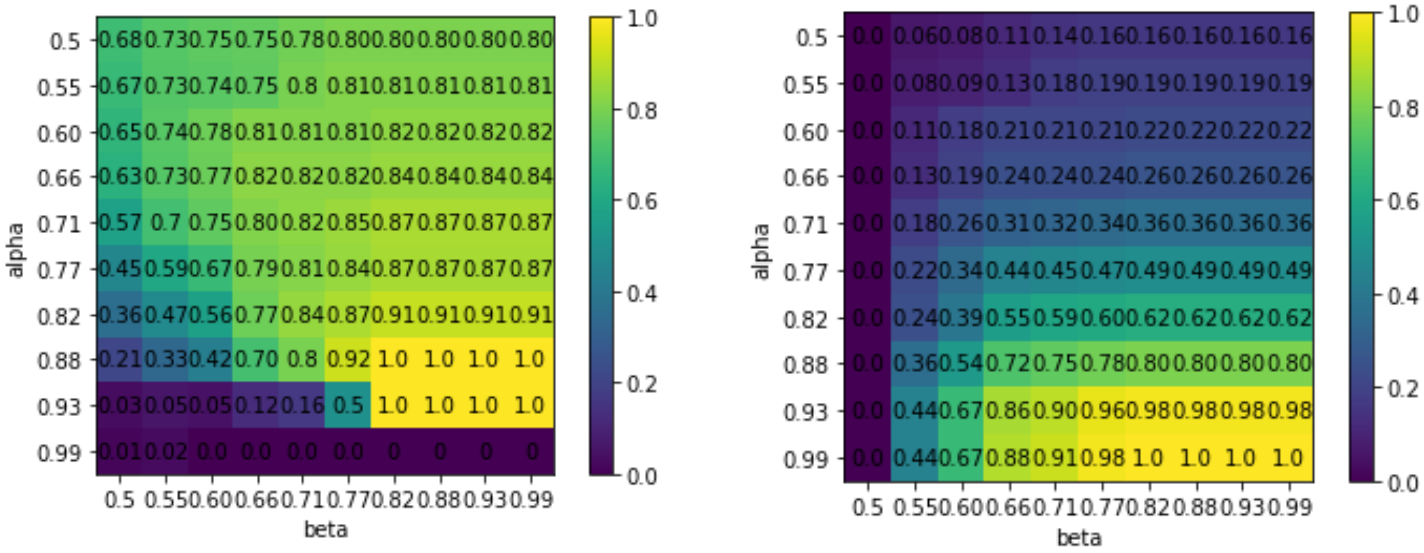


Схема 2 Алгоритм классификации новых писем.

4.3 Оценка качества классификации

Для оценки работоспособности модели и качества алгоритма классификации на личный почтовый ящик на платформе GMail были посланы письма, принадлежащие пользователю “*love-p* “ из Ernon Email Dataset. Для них был запущен полный цикл работы разработанного скрипта. Письма пользователей были разделены на обучающую и тестовую выборку. У писем в тестовой выборке была удалена метка класса. Предсказание классов писем производилось при различных параметрах уверенности классификации *alpha* и *beta*, распределенных на регулярной решетке. Для каждой пары параметров была посчитана метрика *accuracy* на классифицированных письмах, а также процент отказа от классификации. Результаты тестирования приведены на Рисунке 2.



Accuracy

Процент отказа от классификации

Рисунок 2 Результаты тестирование скрипта на пользователе ‘love-p’.

5 Заключение

В результате прохождения производственной практики мною были получен опыт работы с Gmail API, освоены технологии обработки текстовых данных, методы извлечения признаков из текстовых данных и алгоритмы машинного обучения для их классификации. На основе изученного материала был разработан и реализован алгоритм для автоматической классификации писем электронной почты.

6 Источники

1. https://www.kaggle.com/wcukierski/enron-email-dataset
2. https://1c.ru/rus/firm1c/firm1c.htm
3. https://developers.google.com/gmail/api