1. Вступление

В качестве тестового задания было предложено разработать алгоритм классификации данных на наличие рекламной, вредоносной и прочей нежелательной информации (*спама*). Данные представляют собой сообщения из социальных сетей, форумов и т.п. источников с некоторыми дополнительными сведениями. Поскольку данные содержат индикатор спама, для анализа будут использованы различные алгоритмы классификации, такие как: k ближайших соседей, метод опорных векторов, случайный лес. Также подразумевается наличие некоторой меры ошибок 1-го и 2-го рода.

2. Основная часть

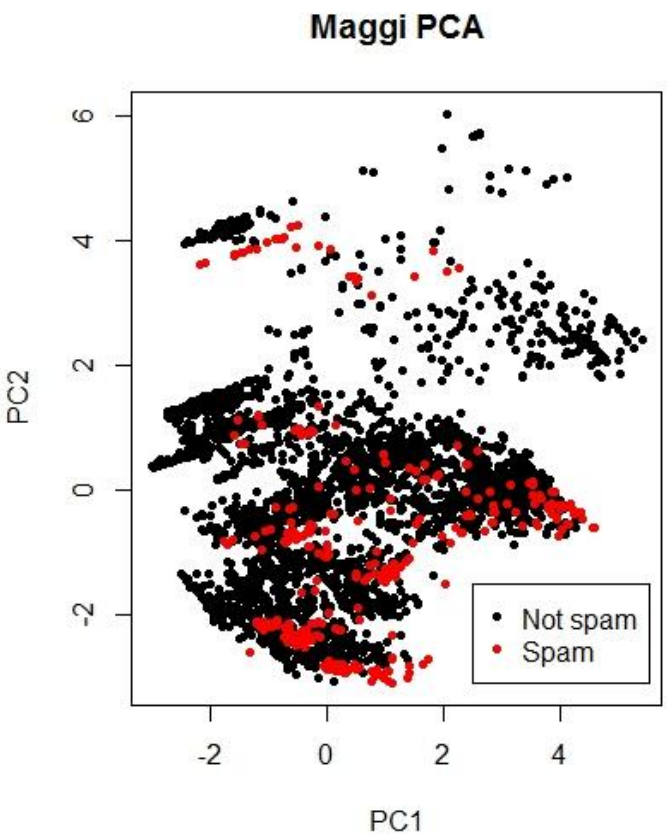
Данные были взяты из 3-ёх тем: Maggi, Raffaello и Nutrilon. В каждой теме было случайно выбрано 20% сообщений, каждое из которых содержит следующие поля:

- Индикатор спама
- Уникальный идентификатор сообщения (в виде URL ссылки)
- Заголовок и текст сообщения
- Настроение сообщения (позитивное, нейтральное или негативное)
- Дата и время размещения данного сообщения
- Информация об авторе сообщения: идентификатор (в виде URL ссылки), имя, тип (пользователь, группа или администратор), время первого появления, время последней активности, количество читателей и URL ссылка на аватар

Все данные были закодированы в матрицу признаков следующим образом:

- Название источника сообщения, а также наличие аватара выделены как отдельные категориальные переменные
- Все категориальные переменные представлены в виде прямого унитарного кода (one-hot encoding)
- Все временны́е переменные переведены в UNIX-время
- Количество читателей рассматриваются в логарифмической шкале*
- Для текстовых данных рассчитаны длины заголовка и самого сообщения, а также индикатор наличия одного из ключевых слов, таких как: *goo, http, insta, …***
- Все слова (наборы символов, разделённые пробелом) в сообщениях являющихся спамом были занесены в словарь. Для каждого сообщения было посчитано количество слов, присутствующих в этом словаре

Практически все данные были корректно описаны, поэтому шаг очистки данных можно пропустить. Следует отметить, что данные являются несбалансированными: количество сообщений, содержащих спам, составляет: 7.86%, 2.93% и 0.49% по каждой теме соответственно.



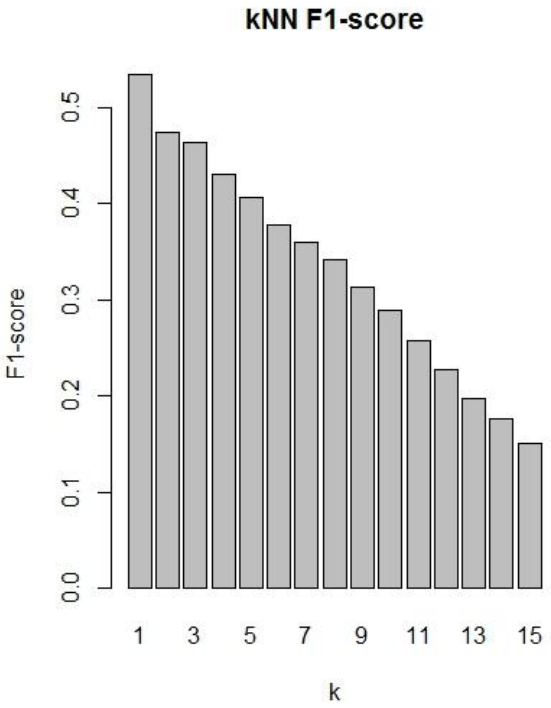
Поскольку классификацию необходимо производить с учётом меры ошибок (ошибки 1-го рода, т.е. классификации обычного сообщения как спама, более значимы) дискриминантные методы (такие как, например, линейный дискриминант Фишера) не смогут решить данную задачу. На графике "**Maggi PCA**" изображены первые две главные компоненты матрицы признаков по теме Maggi. Как видно, данные, скорее всего, не являются линейно отделимыми, следовательно, такие алгоритмы как: логистическая регрессия, перцептрон и т.п. будут плохо работать в данной ситуации. В качестве нелинейных классификаторов рассматриваются: метод k ближайших соседей, метод опорных векторов (с различными типами ядер) и случайный лес.

Данные были случайным образом разбиты на тренировочную и тестовою подвыборки (80% и 20% соответственно). Поскольку данные несбалансированны для оценки качества классификатора была использована F-мера. Тренировочная выборка разбивалась на 75% подвыборку для тренировки алгоритма и 25% для оценки гиперпараметров. Для простоты рассматривались данные только из темы Maggi. Распространение анализа на всю выборку – прямолинейное, но требует бóльших вычислительных ресурсов.

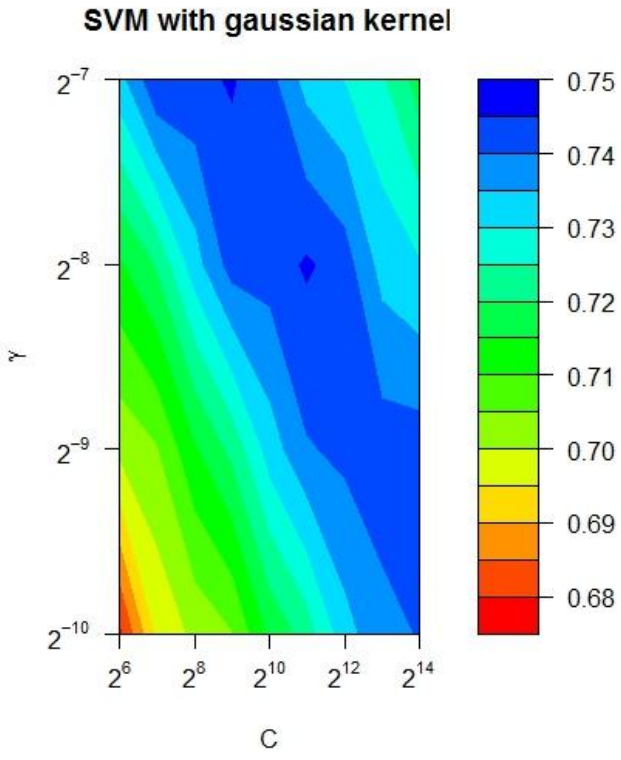
2.1 Подбор гиперпараметров

Во всех методах проводилось несколько одинаковых экспериментов, в которых тренировочная выборка каждый раз случайно разбивалась на две подвыборки: первая из которых (75%) использовалась как обучающая, а вторая (25%) – проверочная и использовалась для подбора гиперпараметров модели. В каждом эксперименте считалась F1-мера для каждого параметра из заранее заданного диапазона. Наилучшими параметрами считались те, которые дают наибольшее значение F1-меры в среднем по всем экспериментам.

В методе k ближайших соседей есть только один гиперпараметр k, который подбирался из диапазона k=1,…, 15. Результаты после усреднения 100 экспериментов изображены на графике “**kNN F1-score**”.



Согласно графику наилучшим значением k является 1, при котором F1 = 0.53. При таком значении k алгоритм не может выдавать вероятностный ответ и, следовательно, плохо подходит для решения данной задачи.

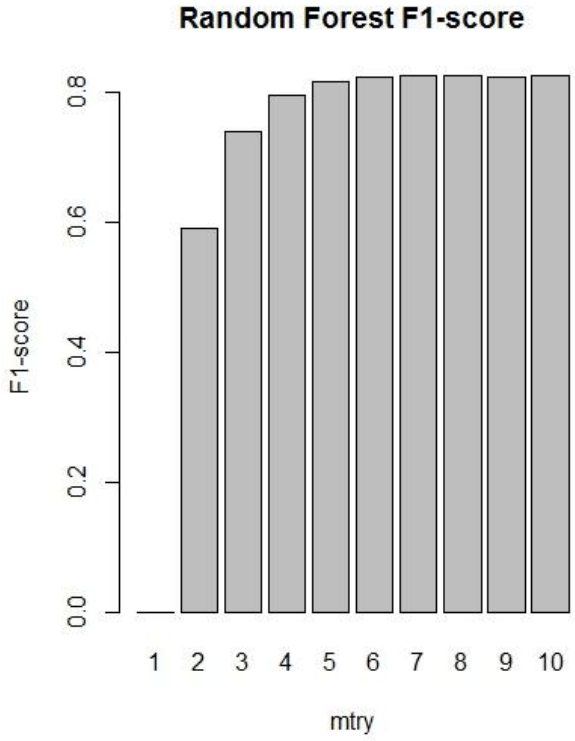


В методе опорных векторов помимо параметров необходимо также подобрать ядро. Для каждого типа ядра параметры подбирались поиском по сетке в логарифмической шкале. Результаты после усреднения 50 экспериментов изображены на графике “**SVM with gaussian kernel**”.

Наилучшими параметрами для данного ядра будут $C = 2^{11}$ и $\gamma = 2^{-8}$, при которых значение F1-меры достигает 0.75. Использование других ядер не дало существенных улучшений: полиномиальное ядро показало примерно такие же результаты, а сигмоидальное

сработало хуже.

В отличие от предыдущих алгоритмов случайный лес не так чувствителен к параметрам. Как видно на графике “**Random forest F1-score**”, оптимальное значение параметра *mtry*=8 лишь немного улучшает F1-меру алгоритма по сравнению с предложенным значением *mtry*=5.



2.2 Результаты тестирования

Для окончательной проверки результатов была использована 20% тестовая выборка, которая не участвовала в процессе обучения. Матрицы неточностей, а также F1-меры каждого алгоритма представлены в таблице. Как и следовало ожидать, случайный лес показал лучшую производительность среди рассмотренных алгоритмов.

Algorithm	**SVM with Gaussian kernel**		**Random Forest**	
Parameters	*cost*=2048, *gamma*=0.00390625		*mtry*=8, *ntree*=500, *nodesize*=1	
Confusion matrix	768	5	771	2
	28	41	18	51
F1-score	**0.804**		**0.836**	

3. Выводы

По результатам анализа нескольких алгоритмов наилучшим оказался случайный лес. Также данный алгоритм не сильно чувствителен к параметрам, что даёт возможность получать хорошие результаты при стандартном выборе параметров. Из недостатков данного алгоритма следует отметить низкую скорость работы, склонность к переобучению на зашумленных данных и большие расходы памяти на хранение модели.

4. Дополнение

* Предположим, что рост кол-ва читателей со временем примерно пропорционален кол-ву читателей на данный момент. Тогда, если $X(t)$ – кол-во читателей в момент t, то X удовлетворяет задаче Коши:

$$\begin{cases} X'(t) = rX(t) \\ X(0) = 1 \end{cases}$$

где r – некоторая константа, отвечающая за скорость роста читателей. Решением данной задачи будет $X(t) = \exp(rt)$. Поскольку в данных присутствуют временны́е переменные, $\ln X$ будет пропорционален им.

** Как правило, вредоносные сообщения содержат ссылки, схожие с «настоящими», например: *goog.le, insta.gram* и т.п.

4.1 Важность признаков (по данным Random forest)

Ещё одним преимуществом алгоритма случайный лес является возможность оценки важности каждого признака. На графике “**Maggi importance**” изображены основные признаки, по которым производилась классификация.