Наивный байесовский классификатор

# Цели работы:

1. практика создания n-грамм;
2. реализовать алгоритм наивного байесовского классификатора;
3. настройка штрафа ошибки с целью минимизации ошибки второго рода;
4. анализ результатов.

# Набор данных

Рассмотрим задачу классификации электронных писем на два класса: спам (spam) и не спам (legit) и соответствующий [набор данных](https://drive.google.com/file/d/1-f_Sr6PXqHyAmz854zbJRDDu9CrI0oRg/). Архив содержит набор данных из писем, каждое из которых состоит из заголовка и текста сообщения. Набор данных уже разбит на 10 частей для перекрёстной проверки. В каждом письме слова закодированы в численном представлении для удобства.

# Задание

n-граммы

Реализуйте преобразование, которое представляет письма в виде разреженного вектора признаков, либо модифицируйте преобразование используемое в соответствующей задаче на Codeforces. Преобразование должно поддерживать [n-граммы](https://ru.wikipedia.org/wiki/N-%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) и учитывать как заголовок, так и содержание письма. Попробуйте различные параметры сглаживания *alpha* и *n* для *n*-грамм. Обратите внимание, что несмотря на ограничения для задачи на Codeforces, параметр сглаживания *alpha* — это не обязательно целое число большее 1. Как правило, оптимальное значение вещественное и сильно близкое к 0. Параметр *n* не стоит брать слишком большим, достаточно проверить *n* = 1, 2, 3*.*

Обучение модели, настройка штрафа ошибки и анализ результатов

Обучите модель на предсказание класса письма. Постройте ROC кривую для обученной модели. Посчитайте точность (accuracy), используя перекрестную проверку (k-fold, где k=10).

Подберите штраф ошибки классификации λlegit такой, чтобы ни одно реальное (legit) сообщение не было классифицировано как спам. Штраф λspam при этом должен быть зафиксирован.

Постройте график зависимости точности от параметра λlegit, где λlegit изменяется от значения по умолчанию (λlegit =λspam ) до найденного в предыдущем пункте значения.