Описание данных:
hepatitis.data - описание симптомов больных гепатитом
Данные выбраны в UCI MLR.
155 объектов, 20 признаков
Присутсвуют как бинарные, так и численные значения (для них применяется шкалирование)

Описание алгоритма:
В данном алгоритме применялся метод на основе генераторов. Сначала данные делились на тестовую и обучающую выборку в соотношении 20% к 80% для 5-fold кросс валидации (5 является оптимальным числом в моем случае — из-за небольшого размера дататсета при $k > 5$ в тестовой выборке могло оказаться совсем мало значений). Далее обучающая выборка делилась на позитивный и негативный контекст. С помощью генераторов для каждого потупающего тестового i-ого объекта проводилось голосование - если пересечение оказывалось только в позитивном контексте, то отдавался 1 голос в пользу класса 1 (здоров), если только в негативном — голос за класс 0 (болен), если же и там, и там — голоса не присваивались.

Модификации:

1. Алгоритм работает на собственном датасете.

2. Пересмотр пространства признаков в имеющемся датасете:

   - Добавление обратного признака к каждому существующему (кроме класса, пол остался так как у женщин и мужчин достаточные признаки для болезни могут отличаться). Это сделано для того, чтобы учитывать не только наличие признаков болезни, но и их отсутствие.
   - С помощью параметра на входе задается процент parameter1%, который в дальнейшем присваивает 1 признаку, если объект обладает более чем parameter1% от числа всех признаков болезни. То есть, если у человека, например 80% признаков, сигнализирующих о болезни, или более, то скорее всего он болен. 0 ставится в противном случае.
   - Учитывая, что в датасете находятся не только главные признаки болезни, но и косвенно к ней относящиеся, в качестве до-

полнительного признака был выбран признак "основной¨ если объект обладает ВСЕМИ признаками из предопределенного списка, ставится 1. В этом списке содержатся те признаки, которыми обладает почти каждый зараженный объект. Список признаков формировался из интернета + анализ негативного контекста в собственном датасете.

3. Одним из недостатков ленивых алгоритмов является длительное время их выполнения. Для того чтобы ускорить свой алгоритм я реализовала распараллеливание всех циклов с помощью foreach и doParallel. Это позволило добиться сравнительно небольшого времени выполнения алгоритма.

4. БОльшая часть признаков имеет бинарную природу, но также в данных присутствуют и численные. Для их преобразования применялось дихотомическое и порядковое шкалирование. Интервалы выбирались индивидуально для каждого признака, основываясь на гистограмме.

Параметры модели:
В качестве параметров были выбраны два значения — parameter1 и parameter2. Первый определяет кол-во признаков, которыми должен обладать объект (см. пункт 2 в модификациях). Второй определяет во сколько раз один класс должен иметь голосов больше, чем другой, для вынесения итогового решения.

Расчет точности:
Основными метриками точности на которых заострялось внимание это accuracy - соотношение верно угаданных классов к числу всех объектов и false positive rate. Дело в том, что в данной задаче я рассмотрела ситуацию когда нам обязательно нужно присвоить класс объекту (конечно, можно было бы просто поставить NaN, если голоса разделились или их соотношение меньше заданного параметра parameter2). Поэтому в любом спорном случае я присваивала худший - если алгоритм сомневается какой класс присваивать объекту - он ставит 0 (болен). Идея такая - нам проще потратить средства и исследовать пациента, классифицированного как болен, чем дать ему умереть, классифицировав его как здоров. При этом моей целью было минимизировать число ошибочно предсказанных пациентов.

Наилучшую точность (accuracy=0.85, fpr=0.153) дали следующие значения параметров: parameter1=0.85, parameter2=1.1