Streszczenie

W poniższym dokumencie przedstawiam wnioski wyciągnięte podczas pisania projektu dot. klasyfikacji z Metod Probabilistycznych w Uczeniu Maszynowym. Celem projektu jest porównanie dwóch algorytmów, naiwnego klasyfikatora bayesowskiego oraz algorytmu regresji logistycznej przy problemie klasyfikacji.

Przygotowanie danych

Na wejściu otrzymujemy zbiór danych postaci $D = \{(x_1^{(i)}, x_2^{(i)}, \ldots, x_9^{(i)}, y^{(i)}) : i = 1, 2, \ldots, m\}$, zatem mamy 9 niezależnych cech opisujących badanie oraz informację, czy badany rak jest łagodny ($y = 2$), czy złośliwy ($y = 4$). Dla ułatwienia będę oznaczał $y = 2$ jako klasę 0, zaś $y = 4$ jako klasę 1 W celu podziału tych danych na zbiory treningowy i testowy, dzielę D na D_0 i D_1 dla danych należących do klas odpowiednio 0 i 1, następnie dzielę obydwa zbiory w proporcji $2 : 1$ i łącząc te części tworzę zbiór treningowy S i testowy T. W ten sposób losowo wybrana dana z S ma takie same prawdopodobieństwo bycia w klasie 0, co losowo wybrana dana z T.

Na potrzeby projektu zauważyłem, że każda z cech jest pewną liczbą całkowitą ze zbioru $\{1, 2, \ldots, 10\}$. W związku z tym potraktowałem te cechy jako zmienne dyskretne, więc nie stosuję w tym przypadku standaryzacji danych. Przy **naiwnym klasyfikatorze bayesowskim** standaryzacja danych i tak nie miałaby sensu, a dla **regresji logistycznej** zakresy tych danych są na tyle małe, że przy liczeniu gradientu nie miało to wpływu na dobranie współczynnika `learning rate`, czy nie utrudnia wyboru hiperparametru λ przy regularyzacji.

Wstępna analiza danych

Regresja logistyczna

Na potrzeby tego projektu zaimplementowałem regresję logistyczną używając do tego klasycznego algorytmu spadku wzdłuż gradientu. Nie zdecydowałem się na żadne optymalizacje, np. **mini−batch**, gdyż model trenował się względnie bardzo szybko: dla $n = 15000$ iteracji (dla tylu epok wyniki już były satysfakcjonujące), na pełnym zbiorze treningowym przebieg algorytmu zajmował ok. 5 sekund

Naiwny klasyfikator Bayesowski

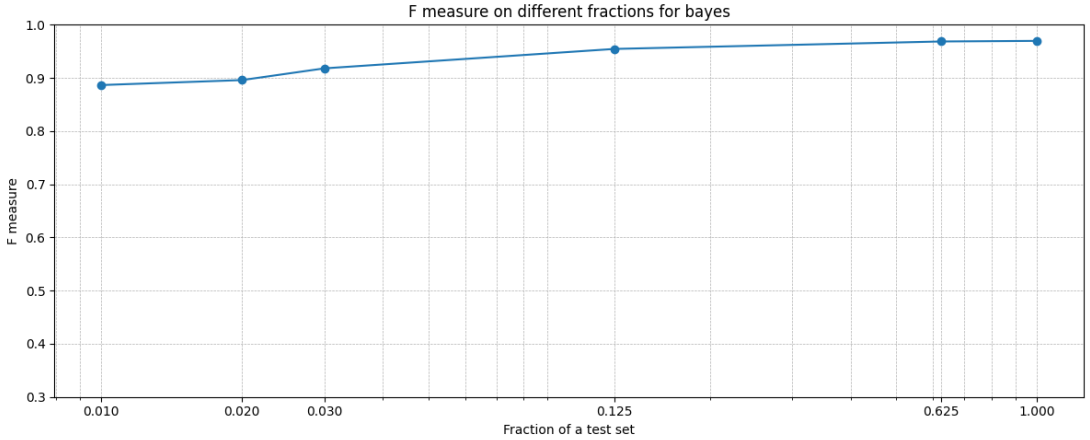
Ciekawą rzeczą, którą zauważyłem analizując proces trenowania naiwnego klasyfikatora bayesowskiego, było to że gdy używałem więcej niż 12.5% zbioru treningowego to im więcej miał danych, tym bardziej pewne decyzje na zbiorze testowym podejmował. Zatem zdecydowaną część wyników szacował albo na 0.00%, albo na 99.99% (decyzje były *troszeczkę* bardziej zrównoważone gdy użyłem `log-prawdopodobieństwa`, ale odpowiedzi nadal były, delikatnie rzecz ujmując, **stanowcze**).

Wyniki

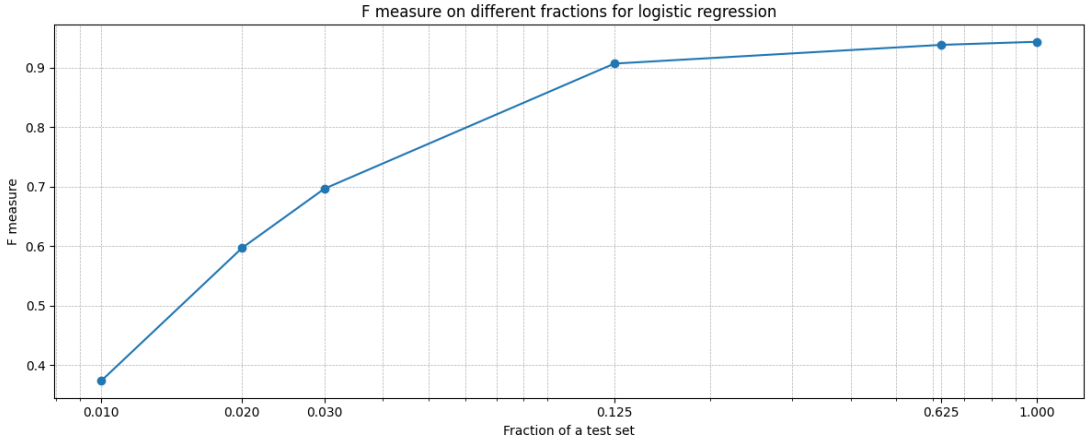
Miarą, której używam do oceny mojego klasyfikatora jest miara F_1, gdyż zależy mi obecnie na modelu który zarówno wykrywa jak najwięcej przypadków i przy okazji unika fałszywych alarmów.

$$F_1(\text{precision}, \text{sensitivity}) = \frac{2 \cdot \text{precision} \cdot \text{sensitivity}}{\text{precision} + \text{sensitivity}}$$

Pytanie na które chciałbym odpowiedzieć, to ile danych wystarczy żeby satysfakcjonująco wytrenować klasyfikator. W tym celu wytrenowałem obydwa modele na różnych frakcjach zbioru treningowego, a wyniki znajdują się poniżej.



Rysunek 1: Miara F_1 dla frakcji przy uczeniu naiwnego klasyfikatora Bayesowskiego



Rysunek 2: Miara F_1 dla frakcji przy uczeniu algorytmem regresji logistycznej

Od początku widzimy bardzo dużą różnicę. Otóż, gdy popatrzymy na skalę miary F_1, naiwny klasyfikator bayesowski od początku radzi sobie dużo lepiej, przy czym już 1% zbioru treningowego pozwala mu na nauczenie się danych na tyle dobrze, żeby otrzymywać średnio miarę 0.89. Jest to wynik do którego regresja logistyczna zbliża się dopiero przy 12.5% zbioru treningowego, a nasz klasyfikator Bayesowski otrzymuje już miary rzędu 0.96