MPUM - Projekt 2

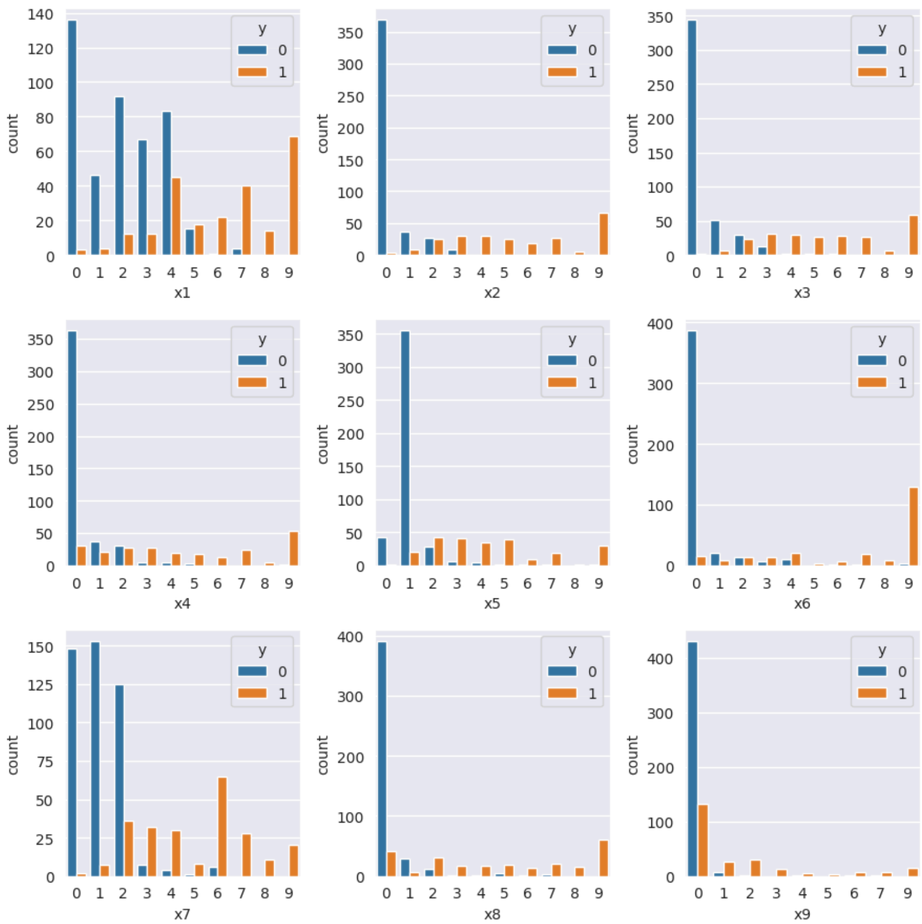
Marcin Wykpis

1. Przygotowanie danych

Na samym poczatku lekko zmodyfikowałem format danych, tak by mogły one zostać wczytane do `DataFrame`'a. Kolumny w przeformatowanym pliku rozdzielone są przez `\t`. Ponadto wszystkie cechy x_i zostały zmapowane w wartości $\{0, 1, ..., 9\}$, a koklumna wynikowa y w $\{0, 1\}$.

2. Analiza danych

W celu przeanalizowania rozkładu cech x_i ezględem kolumny y stworzyłem tak zawny `countplot`. Wykres ten tworzy 2 zestawy 10 słupków: po jednym dla każdej wartości $x_i \in [0, 9], y \in [0, 1]$. Kolumny dla $y = 0$ zostały oznaczone kolorem niebieskim, a pozostałę kolorem pomarańczowym.



3. Podział danych

Dane podzieliłem w stosunku 2 : 1 zbioru treningowego do zbioru testowego.
Zadbałem również o to żeby zarówno dane z pozytywną jak i negatywną diagnozą były podzielone w tym stosunku. Podział danych jest losowany. Zmieniając parametr `seed` uzyskujemy rózne podziały zbioru.

4. Ocena predykcji

Żeby ocenić jak dobrze radzą się moje modele cdecydowałem się użyć funkcji `F-score` z parametrem $\beta = 10$.

$$F_\beta = (1 + \beta^2)\frac{\text{precyzja} \cdot \text{czułość}}{\beta^2 \cdot \text{precyzja} + \text{czułość}}$$

Wybrałem wartość $\beta > 1$, gdyż zależy nam na dokładności (precision) bardziej niż na precyzji (recall). Jest to spowodowane faktem, że wolimy niesłusznie zdiagnozwać pacjenta pozytywnie, podczas gdy brak diagnozy jest niedopuszczalny.

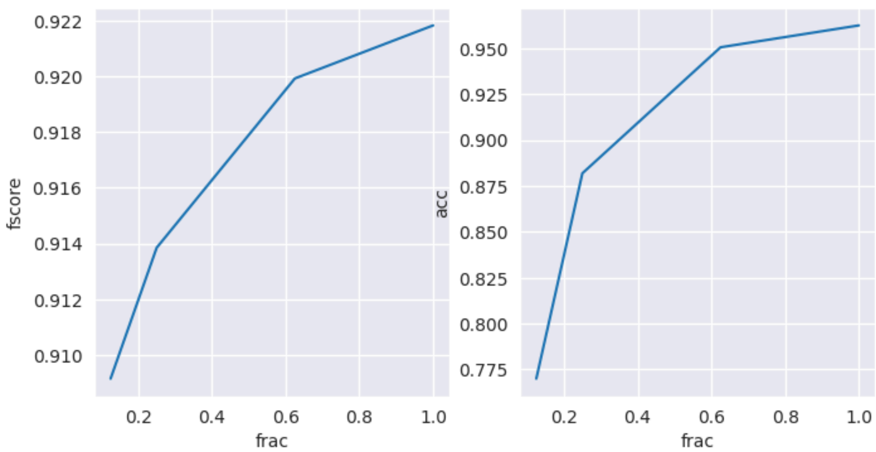
5. Naiwny Bayes

Model ten posiada ten 9 zestawów 10×10 cech zależnych od wartości x_i oraz y oraz 10 cech zależnych jedynie od wartości cechy y.
Cechy te oblcizane są na podstawie utworzonego zbioru treningowany za pomocą nastepujących wzorów:

$$\Phi_{x_j=\mathrm{cx},y=\mathrm{cy}} = \frac{\sum_{i=1}^{m} \mathbf{1}\left[x_j^{(i)} = \mathrm{cx}, y^{(i)} = \mathrm{cy}\right]}{\sum_{i=1}^{m} \mathbf{1}\left[y^{(i)} = \mathrm{cy}\right]}$$

$$\Phi_{y=\mathrm{cy}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\left[y^{(i)} = \mathrm{cy}\right]$$

Krzywe uczenia dla wartości funkcji F-score, oraz dokładność prezentują się następująco. Dla uśrednienia wyniku, dla każdej wielkości danych wytrenowane zostało 100 modelów na różnym podziale danych na treningowe i testowe. Wyniki te zostały bastępnie uśrednione. Modele te zostały wytrenowane na $[0.125, 0.250, 0.625, 1.0]\%$ danych testowych (dla bardzo małej części danych liczba predykcji *true-positive* wynosiła 0, co uniemożliwiło policzenie wartości funkcji f-score).



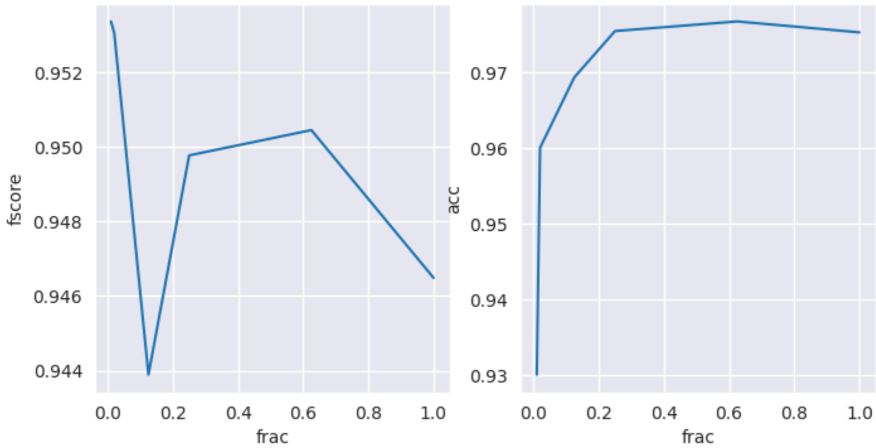
5.1. Wygładzanie Laplace'a

Ten wariant naiwnego bayesa, jest z grubsza wyliczany w ten sam sposób co wcześniej przytoczozny naiwny bayes. Cechy tego modelu prezentują się następująco:

$$\Phi_{x_j=\mathrm{cx},y=\mathrm{cy}} = \frac{1 + \sum_{i=1}^{m} \mathbf{1}\left[x_j^{(i)} = \mathrm{cx}, y^{(i)} = \mathrm{cy}\right]}{2 + \sum_{i=1}^{m} \mathbf{1}\left[y^{(i)} = \mathrm{cy}\right]}$$

$$\Phi_{y=\mathrm{cy}} = \frac{1}{m+2}\left(1 + \sum_{i=1}^{m} \mathbf{1}\left[y^{(i)} = \mathrm{cy}\right]\right)$$

Krzywe uczenia dla wartości funkcji F-score, oraz dokładność prezentują się następująco. Dla uśrednienia wyniku, dla każdej wielkości danych wytrenowane zostało 100 modelów na różnym podziale danych na treningowe i testowe. Wyniki te zostały bastępnie uśrednione. Modele te zostały wytrenowane na $[0.01, 0.02, 0.125, 0.250, 0.625, 1.0]\%$ danych testowych.



6. Regresja Logistyczna

Model ten wykorzystuje regresję logistyczną. Krok w metodzie spadku gradientu wygląda następująco

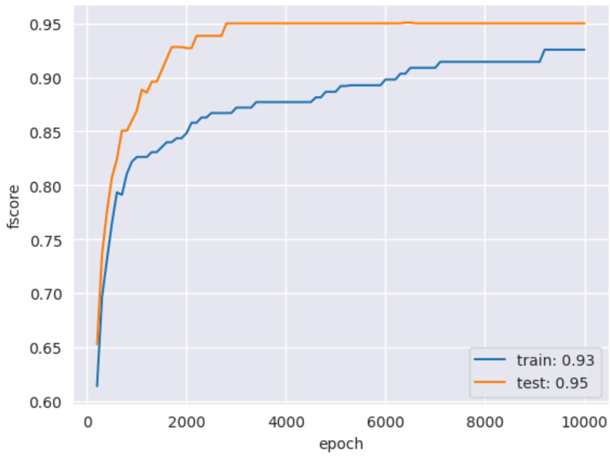
$$\theta_j = \theta_j + \text{step} \cdot (y - h_\theta(X))X_j$$

gdzie

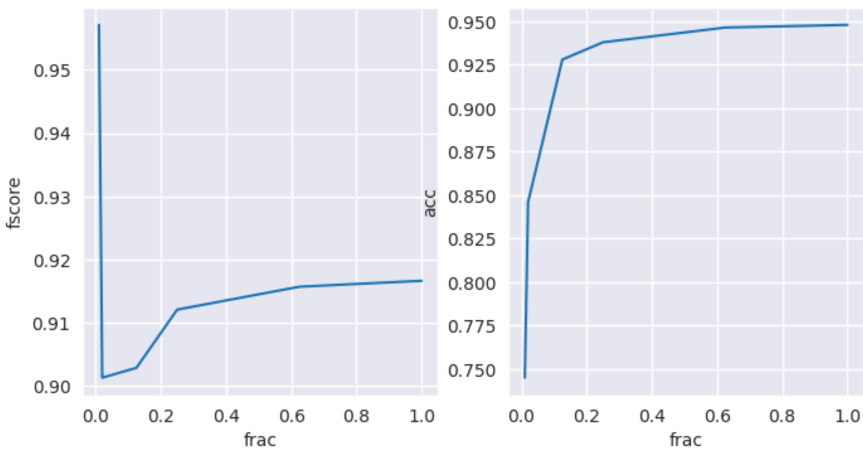
$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}}$$

Model został wytrenowany na 10.000 iteracjach z wartością hiperparametrem `step = 0.001`.

Poniżej znajduje się wykres jak zmieniała się fscore z kolejnymi epochami:



Krzywe uczenia dla wartości funkcji F-score, oraz dokładność prezentują się następująco. Dla uśrednienia wyniku, dla każdej wielkości danych wytrenowane zostało 100 modelów na różnym podziale danych na treningowe i testowe. Wyniki te zostały bastępnie uśrednione. Modele te zostały wytrenowane na $[0.01, 0.02, 0.125, 0.250, 0.625, 1.0]\%$ danych testowych.



7. Wnioski

We wszystkich trzech przypadkach dla krzywej uczenia dokładność oraz F-score zbiega do pewnej wartości. Dla modelu z wygłądzaniem Laplace'a wartość funckji F-score na wykresie skacze, jednak są to różnice rzędu 10^{-3}.

Podobnie jak w artykule *On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes* osattecznie wynik regresji logistycznej „dogonił" a następnie uzyskał przewagę nad modelem Naiwnego Bayesa.

8. Implementacja

Implementację powyższych modeli oraz użyte w raporcie wykresy można znaleźć w repozytorium na githubie: https://github.com/Marwyk2003/mpum_miniproject2