7	1.000000
f_3^1 * f_4	0.934055
f_3 * f_4^2	0.912975
f_3^2 * f_4	0.906840
3	0.867767
feature_3_exp_2	0.847247
feature_3_exp_3	0.802864
f_2 * f_3 * f_4	0.779805
feature_3_exp_4	0.761175
f_2 * f_3^2	0.744624
f_3 * f_4 * f_6	0.734175
feature_3_exp_5	0.725373
f_2^1 * f_3	0.715264
f_3^2 * f_6	0.712603
f_3^1 * f_6	0.669817
f_2 * f_3 * f_6	0.577972
f_2^2 * f_3	0.550614
f_3 * f_6^2	0.508773
f_1^2 * f_3	0.441636
f_0^2 * f_3	0.415215
f_3 * f_5^2	0.383960
f_0 * f_2^2	0.319429
f_0^1 * f_2	0.313455
f_0 * f_2 * f_4	0.300624
4	0.299800
feature_4_exp_2	0.298828
f_0 * f_2 * f_6	0.296200
feature_4_exp_3	0.295995
feature_4_exp_4	0.291756
feature_4_exp_5	0.286567
0	0.276799
f_0^1 * f_6	0.267960
f_0^1 * f_4	0.264925
f_0 * f_4 * f_6	0.256588
feature_0_exp_3	0.253670
f_3 * f_5 * f_6	-0.263398
f_1 * f_3 * f_6	-0.271312
f_2 * f_3 * f_5	-0.283287
f_3^1 * f_5	-0.302550
f_3 * f_4 * f_5	-0.341230
f_3^2 * f_5	-0.365582

Rysunek 1: Korelacje o wartości bezwzględnej większej od 0.25

1 first_miniproject

Podzieliłem dane na zbiory treningowy, walidacyjny, testowy w proporcjach 0.6 : 0.2 : 0.2. Stworzyłem model regresji liniowej używający metody gradientowej dla funkcji straty MSE. Osiągnął on następujące wyniki:
MSE
train $\approx$ 14966.83
validation $\approx$ 7296.47
test $\approx$ 5250.29
Potem wykonałem analizę danych i zauważyłem, że najbardziej skorelowane cechy to kolejno cecha 3, 4 i 0. Eksperymentując chciałem stworzyć model, który przewiduje wartość ostatniej cechy tylko na podstawie

1. cechy 3: dał validation = 19247.740842831452
2. cech 3 i 4 dał validation = 12340.639278231203
3. cech $0, 3, 4,$ [cecha 3] $*$ [cecha 4] dał validation = 5119.956974317123
4. cech $0, 3, 4$ dał validation = 7344.3616800552645

Uzyskiwały one słabe wyniki bo pomyliłem się w indeksowaniu (brałem kolumny o indeksach 3 i 4 z macierzy planowania a tam jest przecież dodana kolumna z jedynkami na początku) oraz zapomniałem uwzględnić kolumny z jedynkami. Po naprawieniu tego błędu 2 ostatnie z tych modeli uzyskały lepszy wynik na zbiorze walidacyjnym od modelu ze wszystkimi cechami. Najlepszy był model nr. 3 osiągając
MSE
train $\approx$ 8314.15
validation $\approx$ 5119.96
test $\approx$ 3676.18
Próbowałem usuwać cechy z modelu 3, żeby zobaczyć jak zmieni się wynik na zbiorze walidacyjnym. Po usunięciu cechy 3 wynik na zbiorze walidacyjnym zmienił się z 5119.959446855926 na 62064.0985378222.

1.1 Początek błędu myślowego

Ciekawe jest, że po usunięciu jeszcze cechy 4 (czyli mamy $(0, [3] * [4])$) wynik na walidacyjnym pogorszył się tylko o około 3000.
Model tylko z cechą $([3 * 4])$ osiągnął 71178.99881227095 na walidacyjnym. Jest to ciekawe dlatego, że wartości tej sztucznej cechy są nawet bardziej skorelowane niż wartości cechy 3.
Najlepszy model po dodaniu regularyzacji $L1$ pogorszył wynik na zbiorze walidacyjnym o około 100 a na testowym polepszył o około 50.
Teraz próbowałem dodawać parametry do najlepszego modelu. Dodawanie parametrów $1, 2, 5, 6$ zmieniało błąd na walidacyjnym o około 20 (dodawałem je po jednym na raz). Dodanie parametru 3 drugi raz oraz dodanie $[3] + [4]$ nic nie zmieniło, co było spodziewane dlatego, że są one liniowo zależne od już istniejących parametrów. Dodanie parametrów $[0] * [3], [0] * [4]$ nie zmieniło wyniku w istotny sposób. Dodanie parametru $[3]^2$ polepszyło wynik na walidacyjnym o około 300 oraz przyspieszyło zbieżność na początkowych kilku epokach. Dodanie $[3]^3$ jeszcze bardziej przyspieszyło zbieżność ale pogorszyło wynik na walidacyjnym o około 500.
Dodałem wszystkie parametry z $[3]^2, [3]^3, [3]^4, [3]^5, [3]^6$ co dało mi najlepszy dotąd wynik na walidacyjnym równy 4798.351894614459. MSE pomiędzy pierwszą a drugą iteracją zmniejsza się wtedy o ponad $2 \cdot 10^6$. Dodawanie cech w postaci $[0]^n, [4]^n$ nie poprawiało wyniku w istotny sposób.

1.2 Odkrycie błędu myślowego

Zorientowałem się, że cechy w postaci $[i]^n, [i] * [j]$ powinienem dodawać przed standaryzacją.

1.3 Korekta

Korekta pogorszyła wynik najlepszego modelu o około 2000 na zbiorze treningowym i około 200 na walidacyjnym.
Dodając cechy w postaci $[3]^i$ dla $i \in \{2, \ldots, 200\}$ uzyskałem wynik na walidacyjnym równy 4368.615954268543.
Dodanie $[0]^n$ oraz $[4]^n$ nie poprawiło wyniku.
Usunąłem potęgi i dodałem cechy $\exp\left(\frac{[3]-\overline{[3]}}{s}\right)$ dla $s \in \{2, \ldots, 10\}$ i otrzymałem błąd 5037.689339105426 na zbiorze walidacyjnym.
Eksperymentując dalej dodałem cechę $([3] * [4])^2$ co przyniosło poprawę błędu na walidacyjnym o około 1000.
Dodawanie cech $([3] * [4])^i$ dla $i > 2$ nie dawało istotniej poprawy.
Zauważyłem swój błąd i zamieniłem cechy $\exp\left(\frac{[3]-\overline{[3]}}{s}\right)$ na $\exp\left(-\frac{\left([3]-\overline{[3]}]\right)^2}{s^2}\right)$ dla $s \in \{1, 100\}$ co dało poprawę do 3099.461563941936. Co więcej mogłem teraz usunąć cechy w postaci $[3]^n$ i dostać błąd 2774.78578867818.
Zastosowanie gaussowskiej funkcji bazowej dla cechy $[3]*[4]$ nie przyniosło istotnej poprawy, podobne efekty funkcja gaussowska dała dla cech $0, 4$. Regularyzacja $L1$ pogorszyła wynik a $L2$ zepsuła go kompletnie (błąd $> 10^6$).

1.4 Pewien problem z MSE

Zauważyłem, że moje MSE od początku miało bład w implementacji co prowadziło do błędnych wyników. Co prawda nie jest to zbyt szkodliwy błąd bo rózniło się o stały czynnik od faktycznego ale unieważnia niestety moje poprzednie wyniki pod względem liczbowym. Nie powinno mieć to chyba wpływu na jakość hipotez. Po naprawieniu MSE najlepsza strata na zbiorze walidacyjnym wynosi 5535.337277219477.

Postanowiłem sprawdzić jakie wyniki otrzymam przy trenowaniu z funkcją Hubera. Poprawiło to MSE na walidacyjnym o około 100 ale kosztowało to 2 razy tyle epok.
Postanowiłem spróbować znaleźć najlepszy regresor używający tylko funkcji wielomianowych. Po dodaniu wszystkich cech wielomianowo zależnych od cech $0, 2, 3, 4$ o stopniu $\leqslant 7$ otrzymałem MSE 4053.091305029673 na zbiorze walidacyjnym. W tym momencie dodanie $[3]^i$ dla $i \in \{2, \ldots, 100\}$ nie poprawia wyniku.
Spróbuję usunąć cechy, których współczynniki w wytrenowanym θ są najmniejsze. Po odcięciu wszystkich cech o współczynnikach z wartością bezwzględną < 7 zostaje nam 47 (było 330) cech a MSE na walidacyjnym wynosi 4257.727541233007.
Po odcięciu cech f o $|f| < 20$ zostało ich 14 a MSE wynosi 4919.087971933042.
Teraz postanowiłem odcinać tak, żeby zostawiać tylko k największych na moduł wartości.

Żeby osiągnąć wynik podobny do tego z 46 cechami wystarcza 31 cech.
W międzyczasie postanowiłem sprawdzić rozwiązanie analityczne. Zaważyłem, że na modelu z 31 cechami daje ono MSE około 1300 a na modelu ze wszystkimi wielomianowymi zależnościami około 46. Może to sugerować, że bardzo słabo zbiegam do minimum.

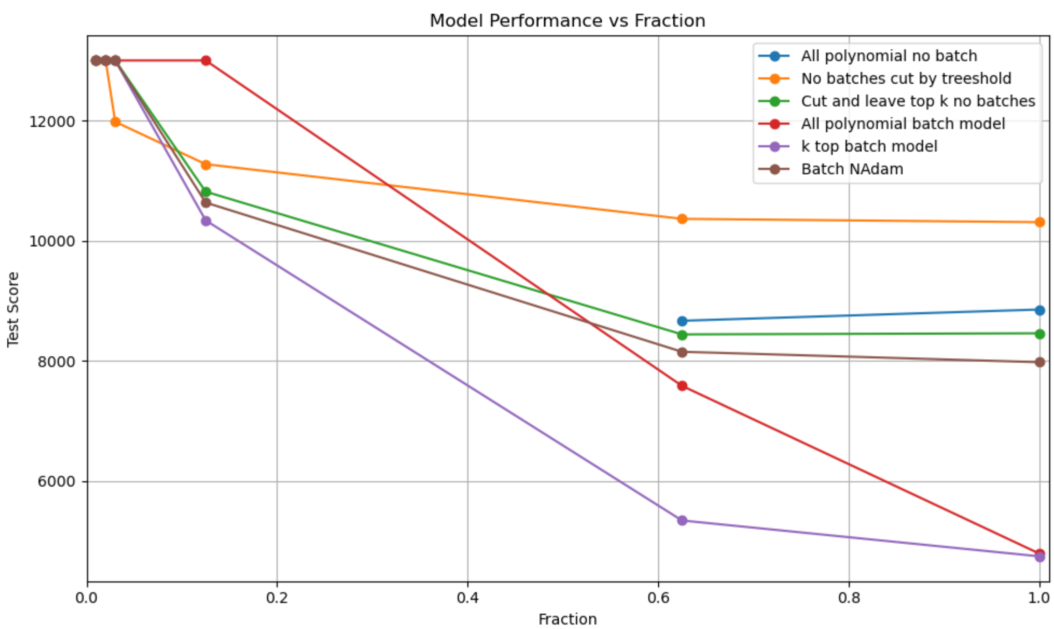
1.5 Inny problem z MSE

Zauważyłem, że moje MSE nadal jest niepoprawne. Domnażałem je dotąd razy $\frac{1}{2}$. Podobnie jak poprzednio nie ma to wpływu na jakość hipotez.

W celu poprawy zbieżności zaimplementowałem regresję opartą na Batchach i uzyskałem wynik 3661.2131984696985 na modelu ze wszystkimi wielomianowymi cechami. Po zwiększeniu liczby epok do 10000 udało mi się otrzymać 2698.151761980878 na zbiorze walidacyjnym.
Po zwiększeniu stopnia wielomianu do 8 i liczby epok do $4 \cdot 10^6$ oraz zaimplementowania learning_rate decay uzyskałem 1996.0439283002372 na walidacyjnym.
Postanowiłem przetestować regularyzację $L2$ na liczbie epok $8 \cdot 10^6$ i parametrze dobranym na podstawie rozwiązania analitycznego na zbiorze walidacyjnym. Znacząco przyspieszyło zbieżność. Po $1,7 \cdot 10^6$ epokach miałem już lepszy wynik niż bez tego po $8 \cdot 10^6$
Równolegle spróbowałem zastosować optymalizator NAdam, żeby zobaczyć czy przyspieszy on zbieżność ale nie uzyskałem satysfakcjonujących rezultatów. NAdam zbiegał zdecydowanie wolniej. Uznałem, że przetestuję włączanie go dopiero po którejś epoce oraz przetestuję go razem z regularyzacją $L2$. Nie poprawiło to wyniku.

Najlepszy wynik przy metodach gradientowych liczył się około 5 min.
Regularyzacja z siecią elastyczną nie była sprawdzana na maksymalnej liczbie epok ale na 2000 epokach nie dawała obiecujących wyników.

1.6 Krzywe uczenia się



Załączam plik regression.sync.ipynb z większością modeli, które sprawdzałem oraz
plik regression_cleaned_up.sync.ipynb z najbardziej obiecujacymi.
Załączam też plik konkurs.py, za pomocą którego udało mi się uzyskać 2.4568885385259764 RMSE.