Explication Contrefactuelle du TCGA

Lecture des revues de littératures sur l'explication contrefactuelle.

Prise en main du TCGA (~10 000 samples, 20 000 features) et codage d'un MLP avec trois couches cachées : 2000 - 200 - 20 -1 avec classification binaire.
Caractéristiques :
BatchNorm, Dropout = 0.02,
SGD with momentum = 0.5,
Learning rate = 0.0003,
L1 penalized loss with lambda = 0.00001,
Batch Size = 64,
Epochs = 300,
Leaky ReLU with negative slope 0.01
Dataset split : 70 training / 30 validation.
Accuracy = 98.56

Sélection de trois types de modèles d'optimisation :
Optimisation directe avec descente de gradient.
VAEs : optimisation sur espace latent ou sphère de rayon croissant
cWGAN-GP

Articles associés :

Sur les VAE :
- C-CHVAE : https://arxiv.org/pdf/1910.09398- VCNet : https://arxiv.org/pdf/2212.10847-
TABCF : https://arxiv.org/pdf/2410.10463
- SCD :https://arxiv.org/pdf/2312.13616

Sur les GAN :
- CTGAN : https://arxiv.org/pdf/2101.10123
- FCEGAN : https://arxiv.org/pdf/2502.17613
- tabGAN : https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3035146
- PCATTGAN : https://arxiv.org/pdf/2003.11323

En optimisation classique (otpimisation sans réduction de dimension):
- DiCE : https://arxiv.org/pdf/1905.07697
- Wachter : https://arxiv.org/pdf/1711.00399
- DiPACE :
https://www.scitepress.org/Papers/2025/132191/132191.pdf#:~:text=The%20primary%20goal%20of%20the,of%20terms%20for%20each%20quality

Première implémentation de DiCE
Possibilité d'adapter le GAN d'Alice

14/03

Objectif : Mettre en place des métriques et les évaluer avec DiCE :
-Validité : y(cf)-y(f) (1=perfect)
-sparsité : 1-(n(changed)-n(total_features)) (1=perfect)
-proximité : L1 distance of features weighted by MAD
-diversité : relative distance of CFs weighted by MAD
-sparse diversité : 1-(n(intersection)/n(union)) (1=perfect)
-adversarial attack (à voir plus tard) (growing sphere sampled and ratio classified with MLP)
-Plausibilité (plutôt garanti avec VAEs et GANs)

DiCE generation method :

```
counterfactuals = exp.generate_counterfactuals(
        factual_instance,
        total_CFs=10,
        desired_class="opposite",
        diversity_weight=1,
        proximity_weight=1,
        sparsity_weight=1
)
```

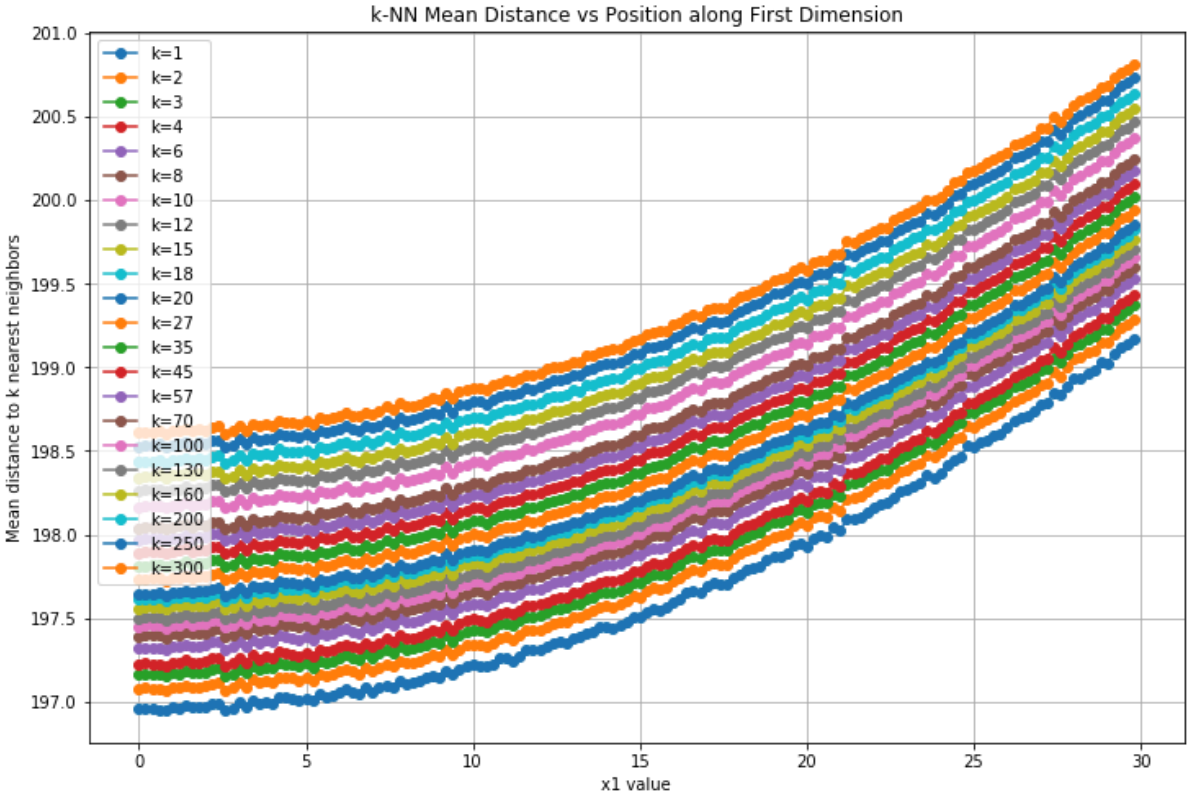
VANILA DiCE :
val : 1
spars : 0.9996
prox : 0.0484
div : 970
sparse div : 0.6576

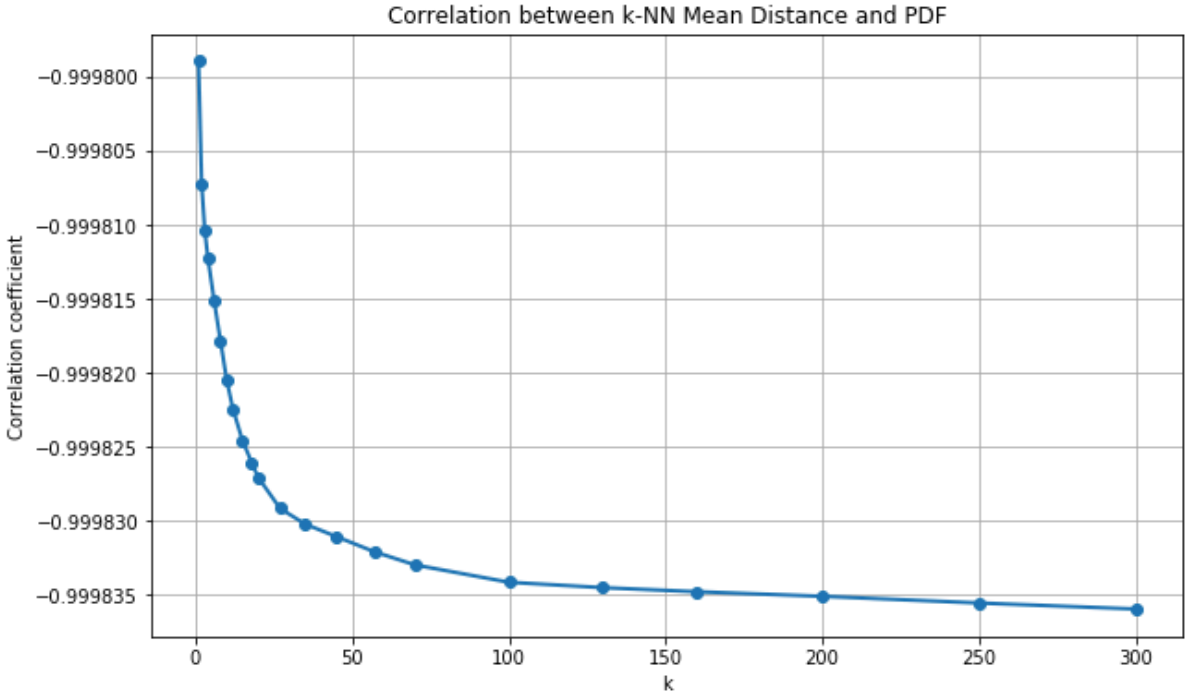
NODIV DiCE :
val : 1
spars : 0.9996
prox : 0.0429
div : 675
sparse div : 0.5706

SPARSITY = 0.5 :
val : 1
spars : 0.9997
prox : 0.0348
div : 577
sparse div : 0.5974

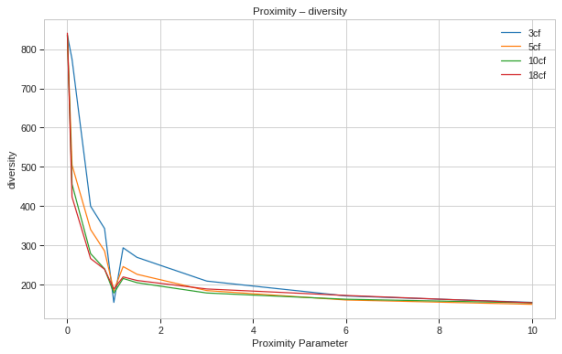
PROX = 0.5 :
val : 1
spars : 0.9997
prox : 0.0373

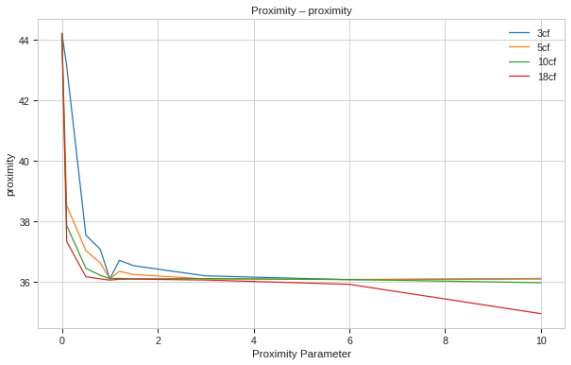
div : 489
sparse div : 0.6098

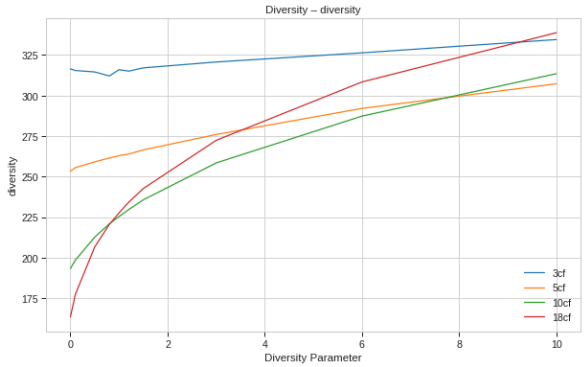




process :
Premiers tests : sparsity_weight + dataset non normalisé => sparsité calcultée par norme L1 dans la loss : features à variance faible effacées = output très sparse et aberrants.
=> pas de calcul post hoc







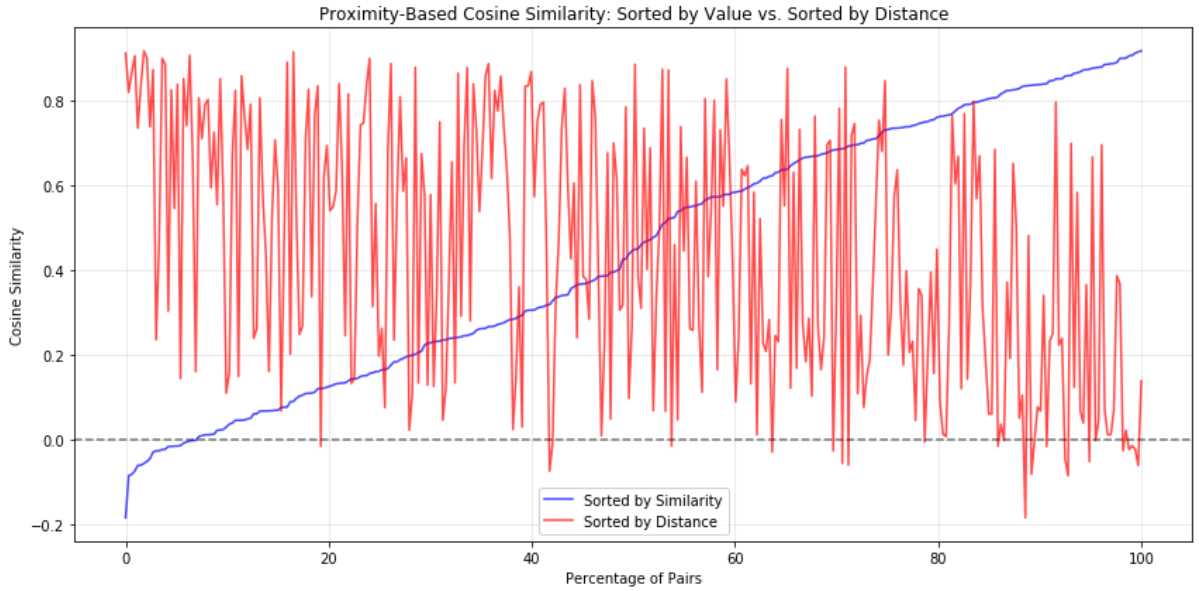
réécriture des fonctions, sur dataset normalisé.
Classements : min(mad, pourcentile) (au début scores très bas car exclusion de toutes les valeurs au delà du seuil)
Tentative de reclassement dynamique des features à chaque revert à partir de la position de chaque feature.

Analyse 3_cf :
ascendant, marche mieux bien que descendant.
plus performant : random
desc : spars = 720
asc : spars = 670
rand : spars = 660
l'algo ne dégrade pas la validité.

le paramètre de sparsité a une incidence négligeable sur le nombre de features changées (+-0.5)
La sparsité rapproche les cf de la frontières (de 5000 à 600/700 en L1) / la proximité passe de 36 à 3-7
Les explications contrefactuelles on cependant du mal à s'approcher des contrefactuels réels avec une L1 à qui passe de 8200 à 9800 avec l'algo post-hoc.

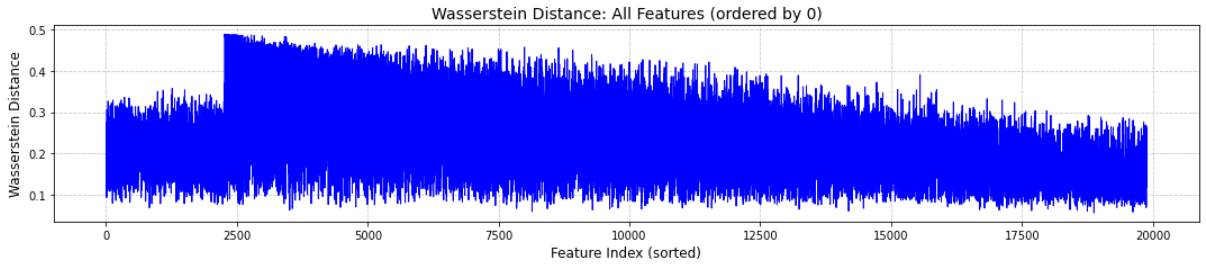
L’algo post hoc augmente cependant la diversité quelque soit la méthode employée, qui passe de 300 à 500.
La sparse diversité est autour de 0.7 pour toutes les méthodes.

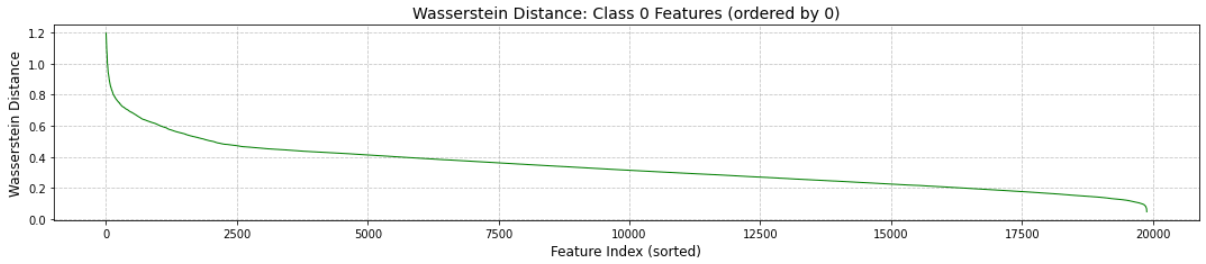
ci-dessous : similarité des vecteurs liant les points les plus proches.

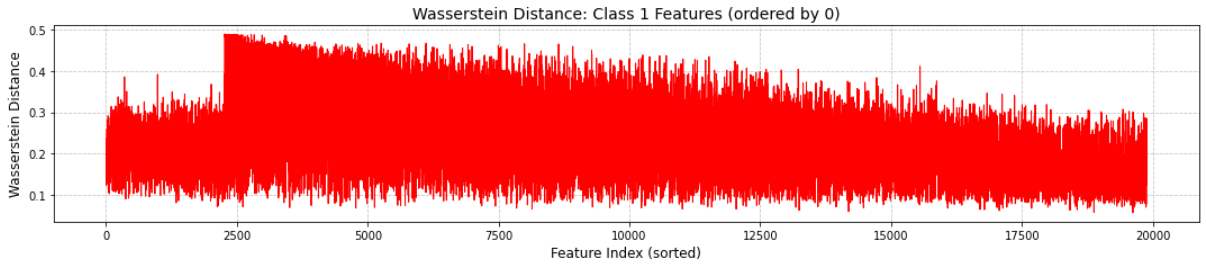


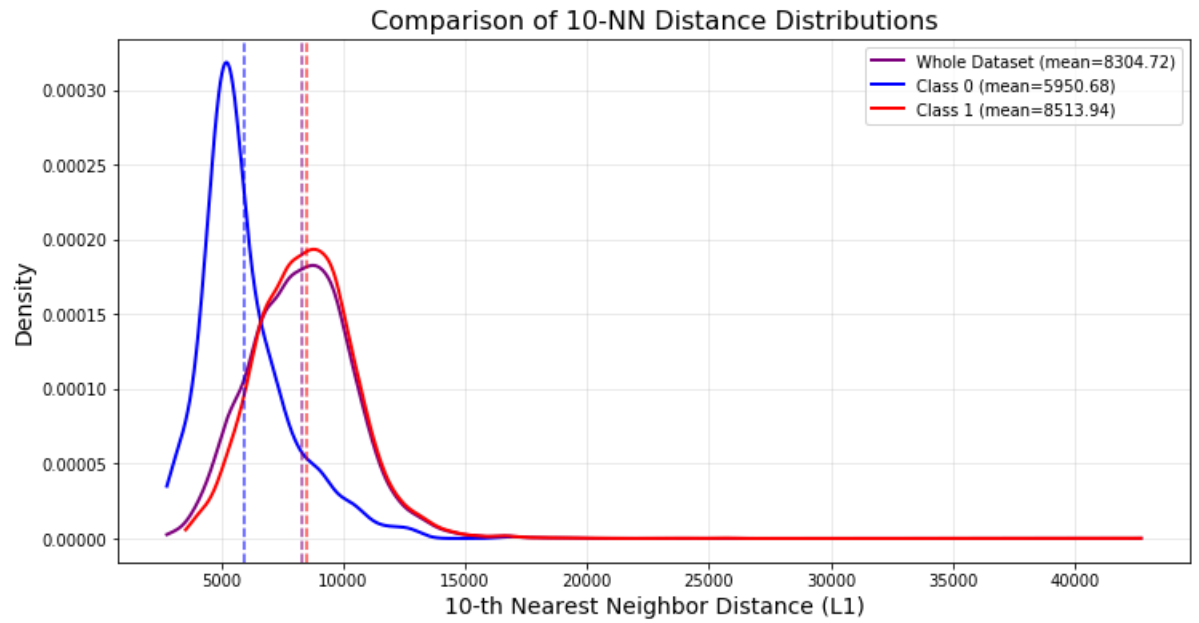
On observe qu’il n’y a pas de tendance claire qui pourrait faire apparaître une forme de courbure dans les plus proches voisins de classe 0 sur l’ensemble des features. La quantité de features est peut-être responsable de ce bruit.
On regarde d’autre part la distance de wasserstein des features par rapport à la moyenne des distibutions. On observe qu’il y a environ 2500 features hors de la distribution moyenne qui caractérrisent la classe 0. Elles sont ici rangées par valeur de distance décroissante selon les samples de la classe 0.

Cette information peut être intéressante pour évaluer d’une part la courbure sur la restriction de ces features, d’autre part pour faire du calcul brute force pour restreindre l’explosion combinatoire.
Etant donné que la distribution des 10-NN est plus à gauche pour la classe 0 alors même qu’il y a moins de points, cela tend à montrer que l’expression des tissus sains est assez similaire tandis que les cancers entoureraient les points sains.

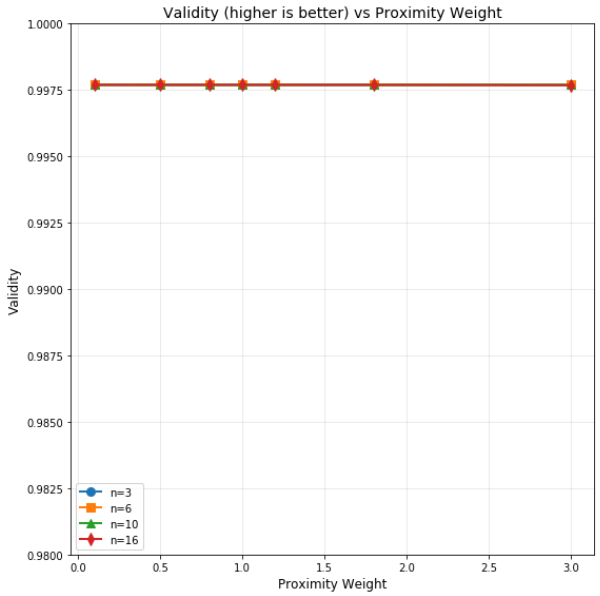


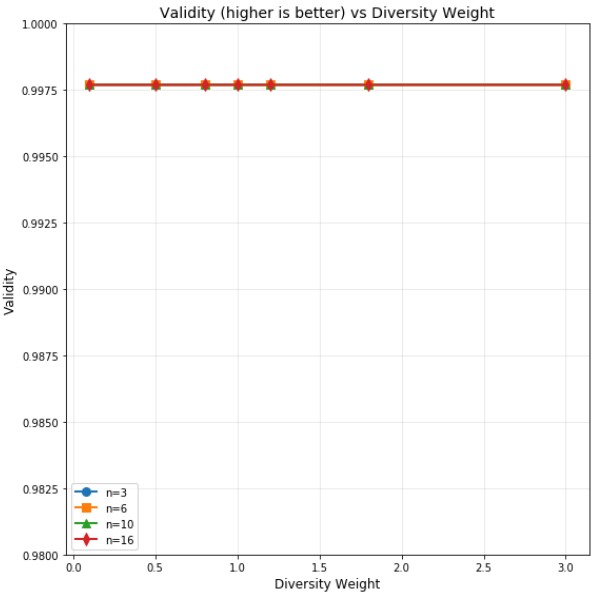


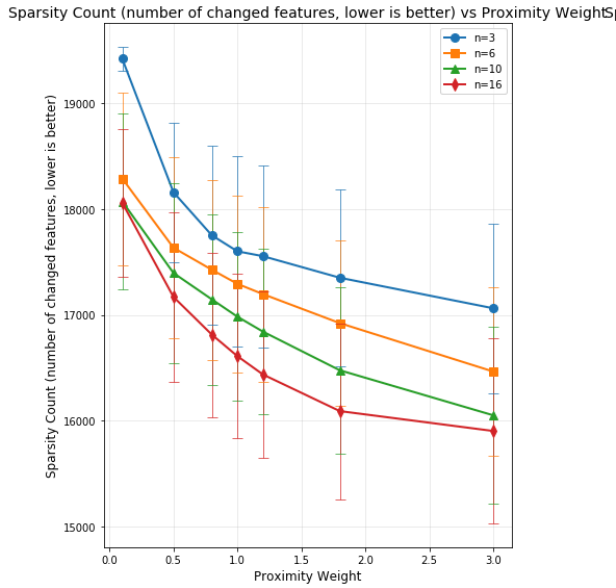


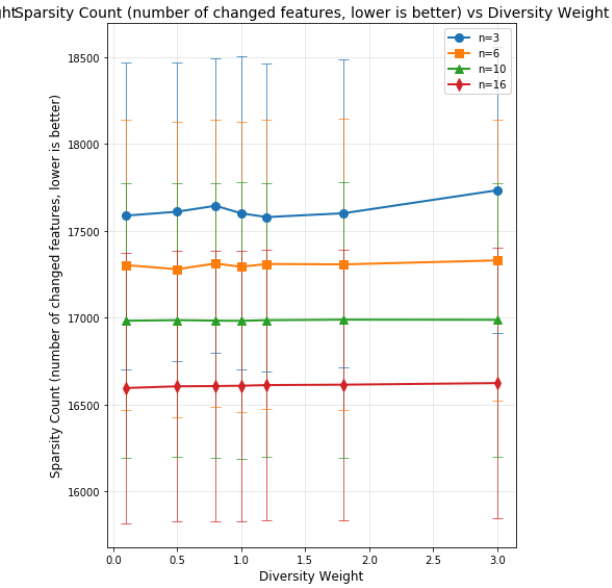


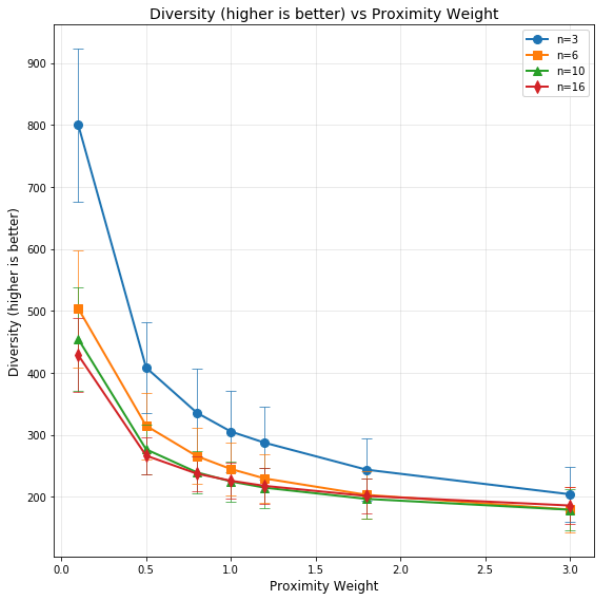
Evaluation des contrefactuels :

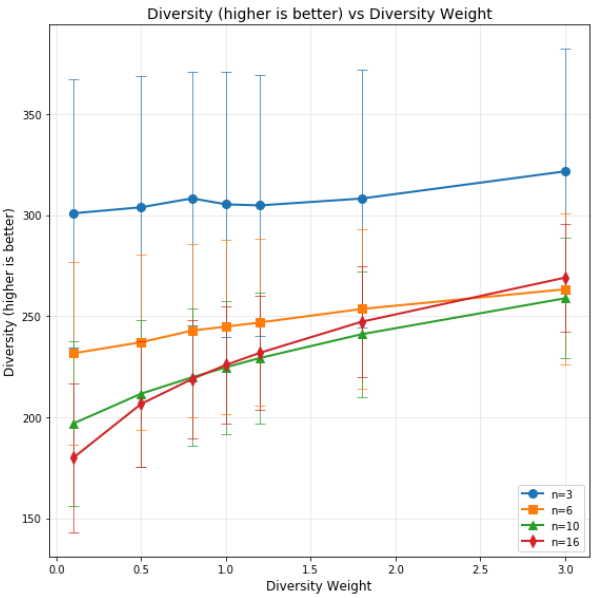

Validity (higher is better) vs Proximity Weight
Validity
Proximity Weight
n=3
n=6
n=10
n=16

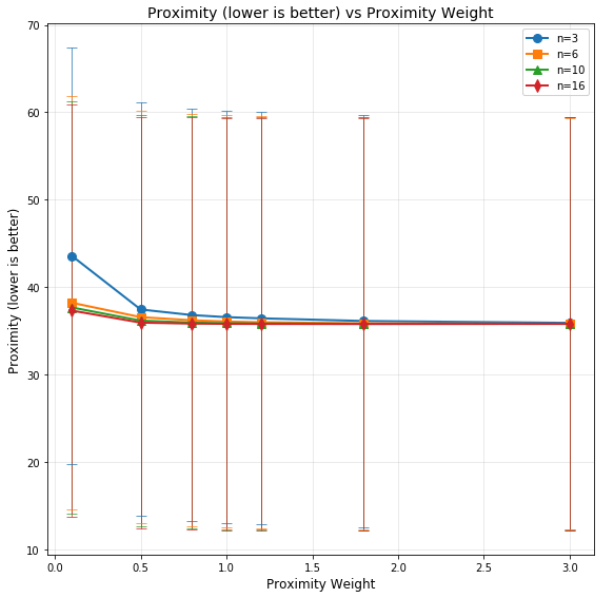

Validity (higher is better) vs Diversity Weight
Validity
Diversity Weight
n=3
n=6
n=10
n=16

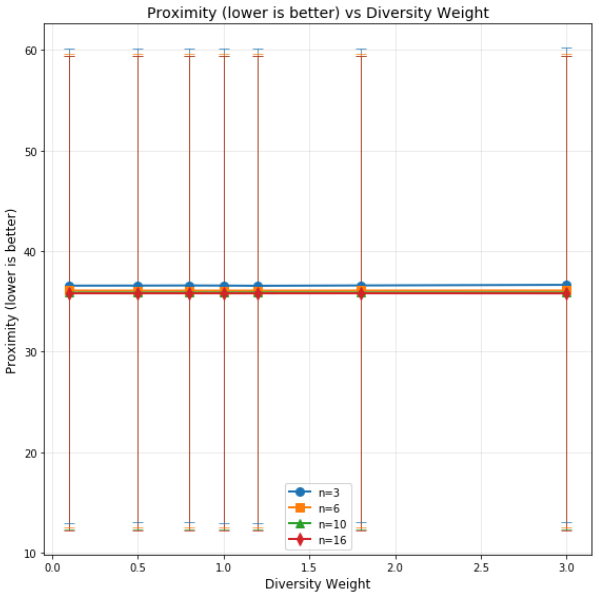

Sparsity Count (number of changed features, lower is better) vs Proximity Weight
Sparsity Count (number of changed features, lower is better)
Proximity Weight
n=3
n=6
n=10
n=16

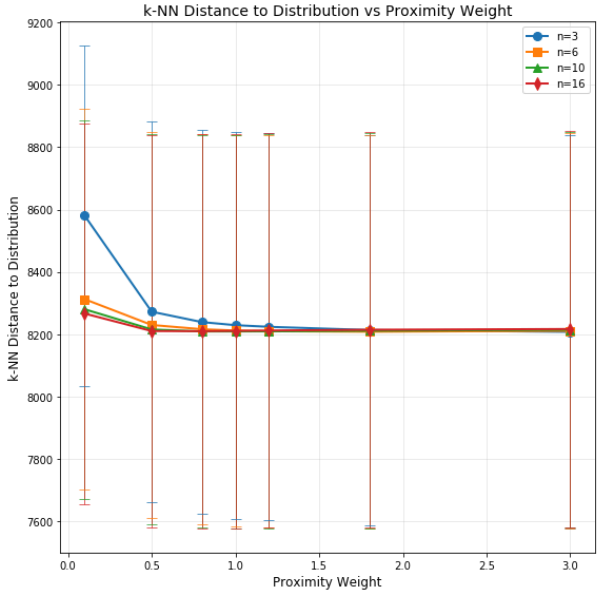

Sparsity Count (number of changed features, lower is better) vs Diversity Weight
Sparsity Count (number of changed features, lower is better)
Diversity Weight
n=3
n=6
n=10
n=16

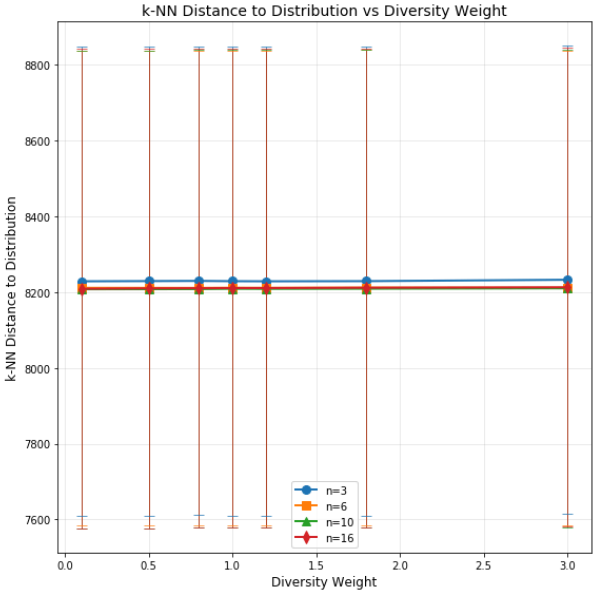

Diversity (higher is better) vs Proximity Weight
Diversity (higher is better)
Proximity Weight
n=3
n=6
n=10
n=16

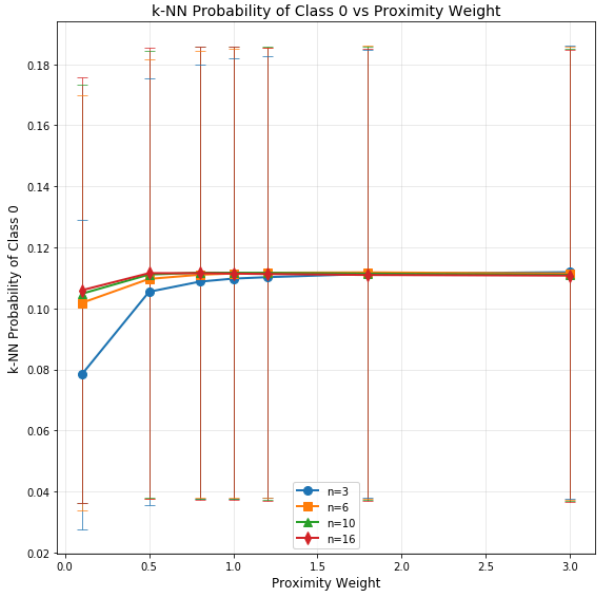

Diversity (higher is better) vs Diversity Weight
Diversity (higher is better)
Diversity Weight
n=3
n=6
n=10
n=16

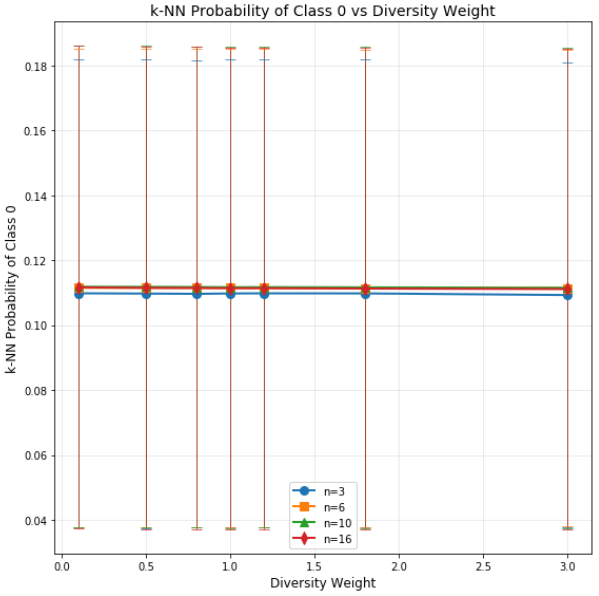

Proximity (lower is better) vs Proximity Weight
Proximity (lower is better)
Proximity Weight
n=3
n=6
n=10
n=16


Proximity (lower is better) vs Diversity Weight
Proximity (lower is better)
Diversity Weight
n=3
n=6
n=10
n=16

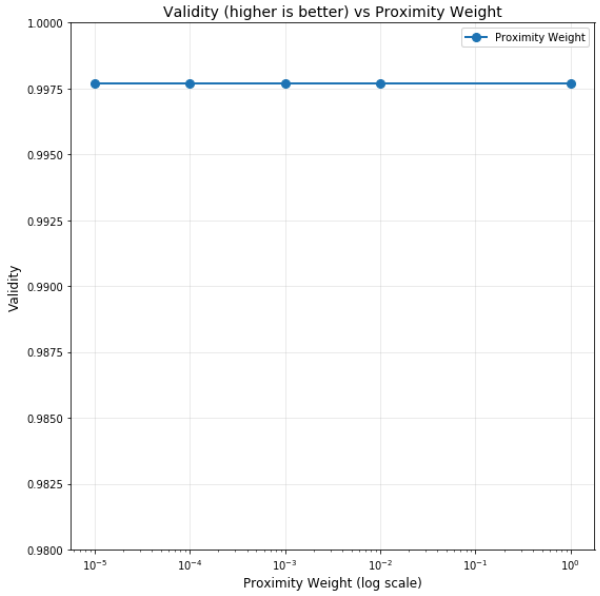

k-NN Distance to Distribution vs Proximity Weight
k-NN Distance to Distribution
Proximity Weight
n=3
n=6
n=10
n=16

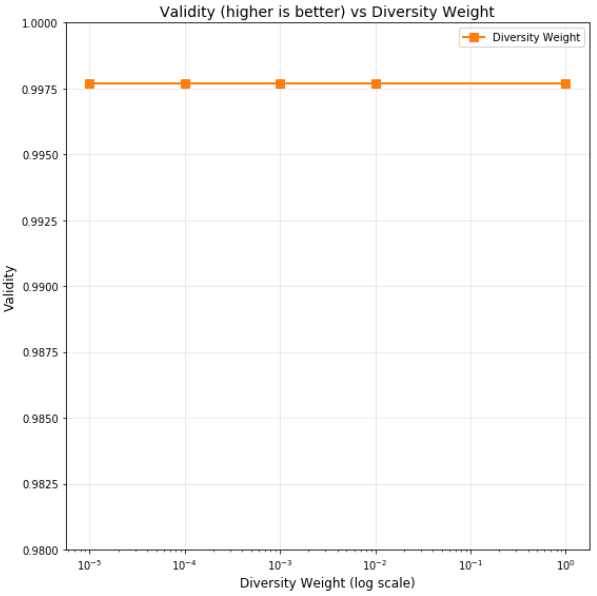

k-NN Distance to Distribution vs Diversity Weight
k-NN Distance to Distribution
Diversity Weight
n=3
n=6
n=10
n=16

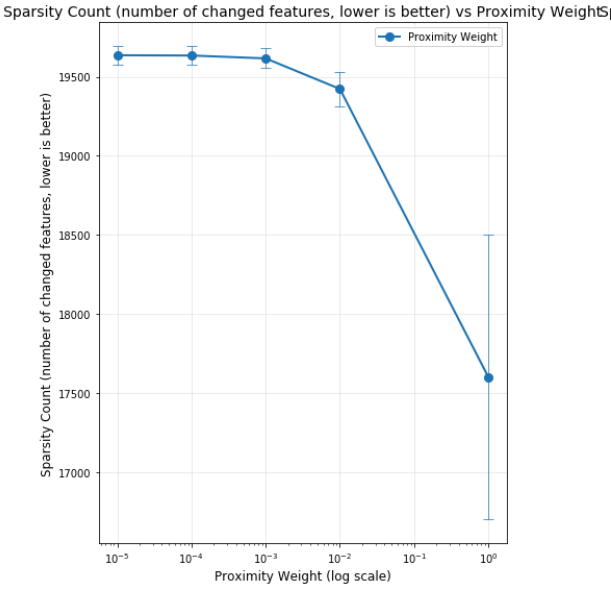

k-NN Probability of Class 0 vs Proximity Weight
k-NN Probability of Class 0
Proximity Weight
n=3
n=6
n=10
n=16

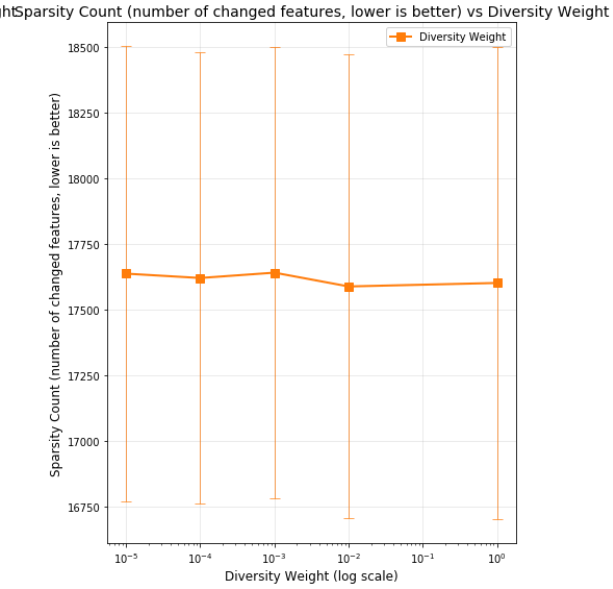

k-NN Probability of Class 0 vs Diversity Weight
k-NN Probability of Class 0
Diversity Weight
n=3
n=6
n=10
n=16

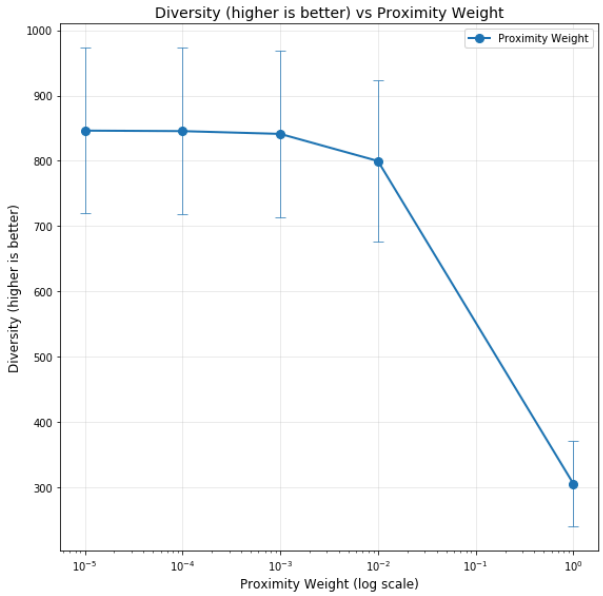
pour des basses valuers de p et d :

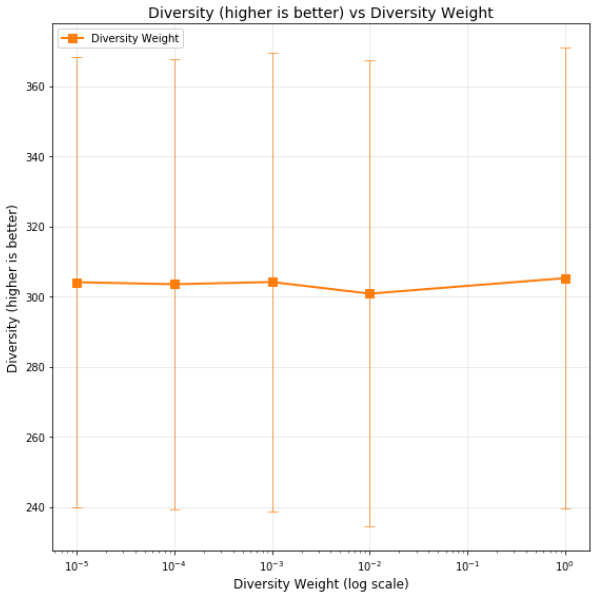


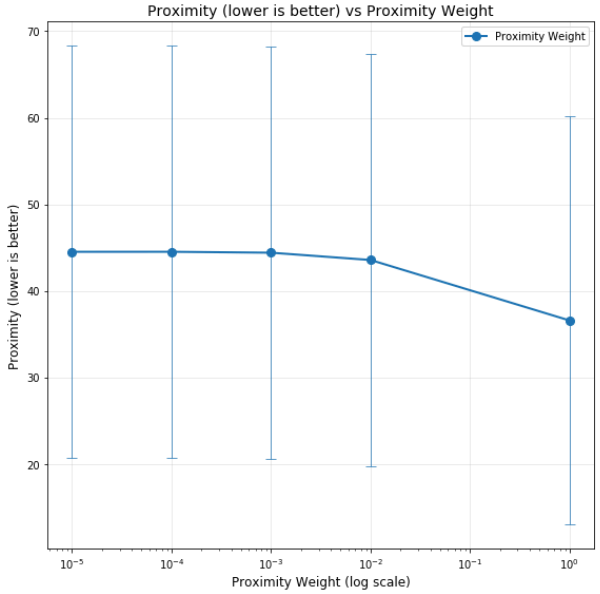


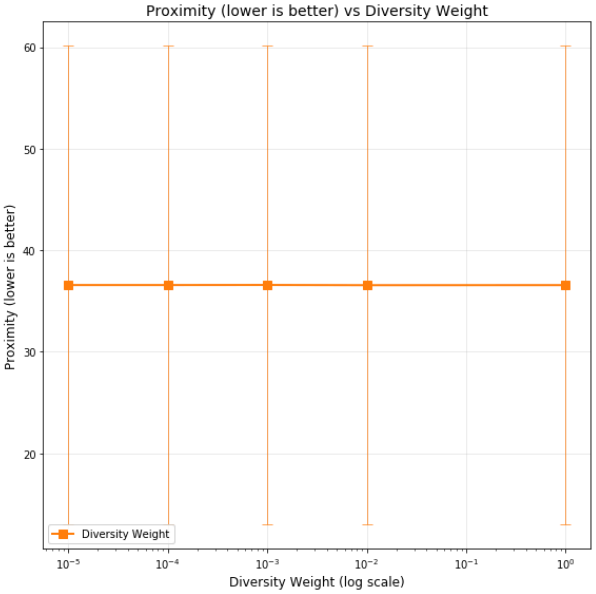


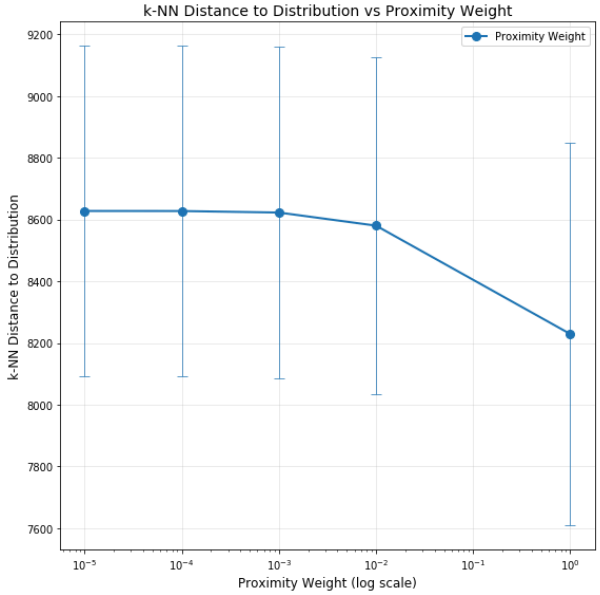


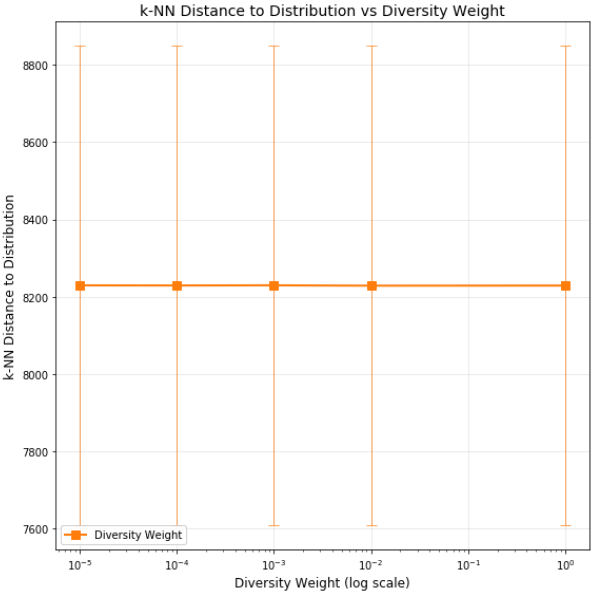

Diversity (higher is better) vs Proximity Weight
Proximity Weight
Diversity (higher is better)
Proximity Weight (log scale)

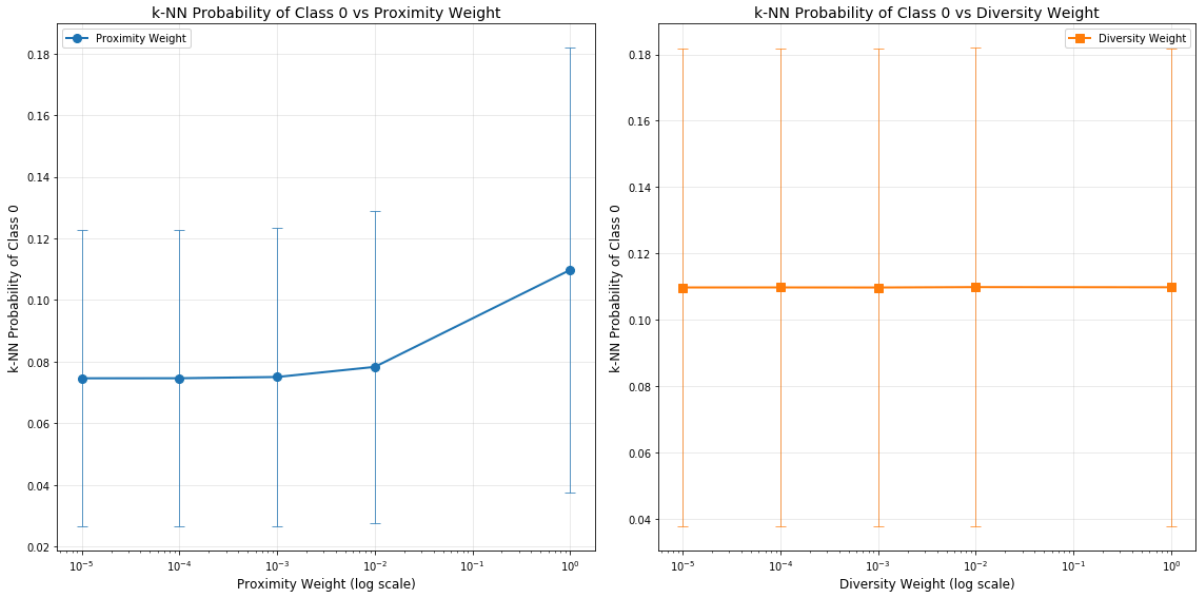

Diversity (higher is better) vs Diversity Weight
Diversity Weight
Diversity (higher is better)
Diversity Weight (log scale)


Proximity (lower is better) vs Proximity Weight
Proximity Weight
Proximity (lower is better)
Proximity Weight (log scale)

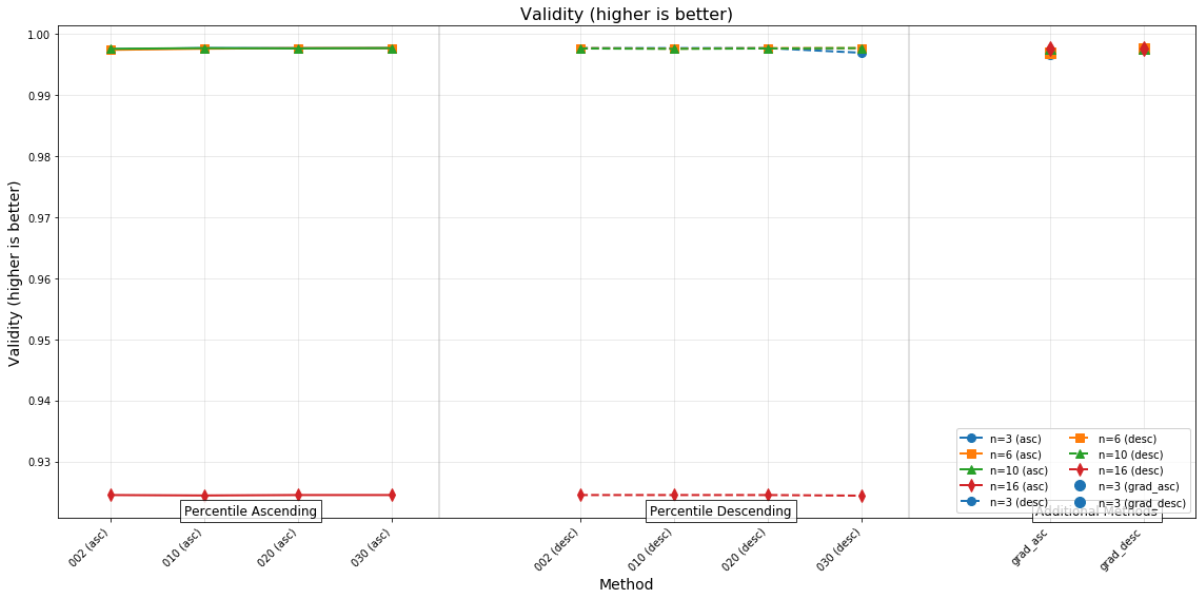

Proximity (lower is better) vs Diversity Weight
Diversity Weight
Proximity (lower is better)
Diversity Weight (log scale)

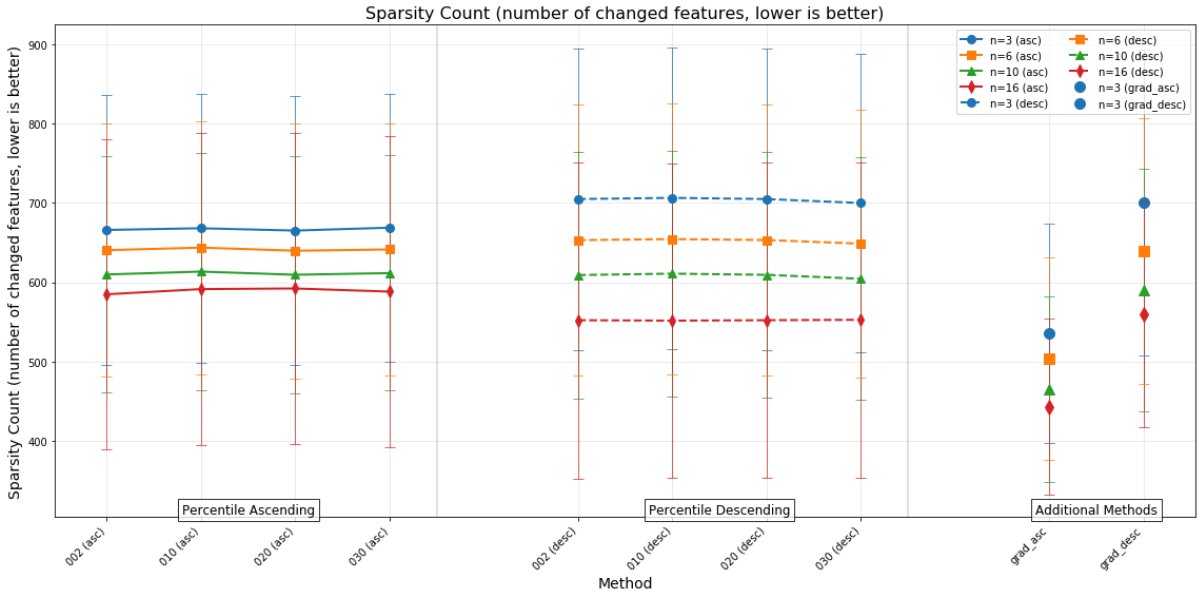

k-NN Distance to Distribution vs Proximity Weight
Proximity Weight
k-NN Distance to Distribution
Proximity Weight (log scale)

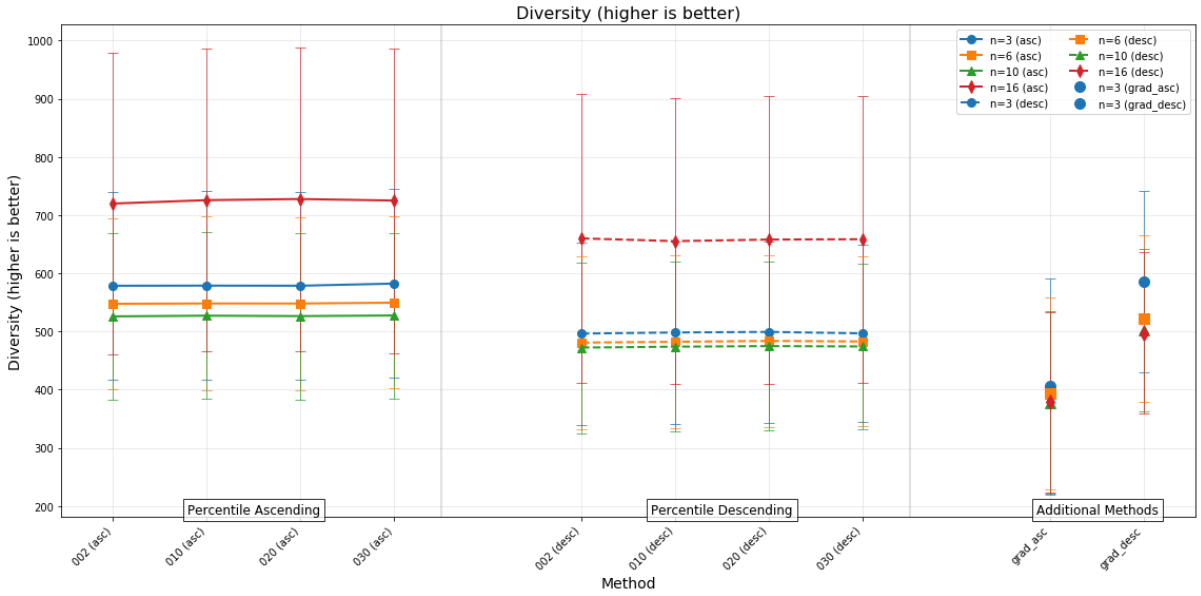

k-NN Distance to Distribution vs Diversity Weight
Diversity Weight
k-NN Distance to Distribution
Diversity Weight (log scale)

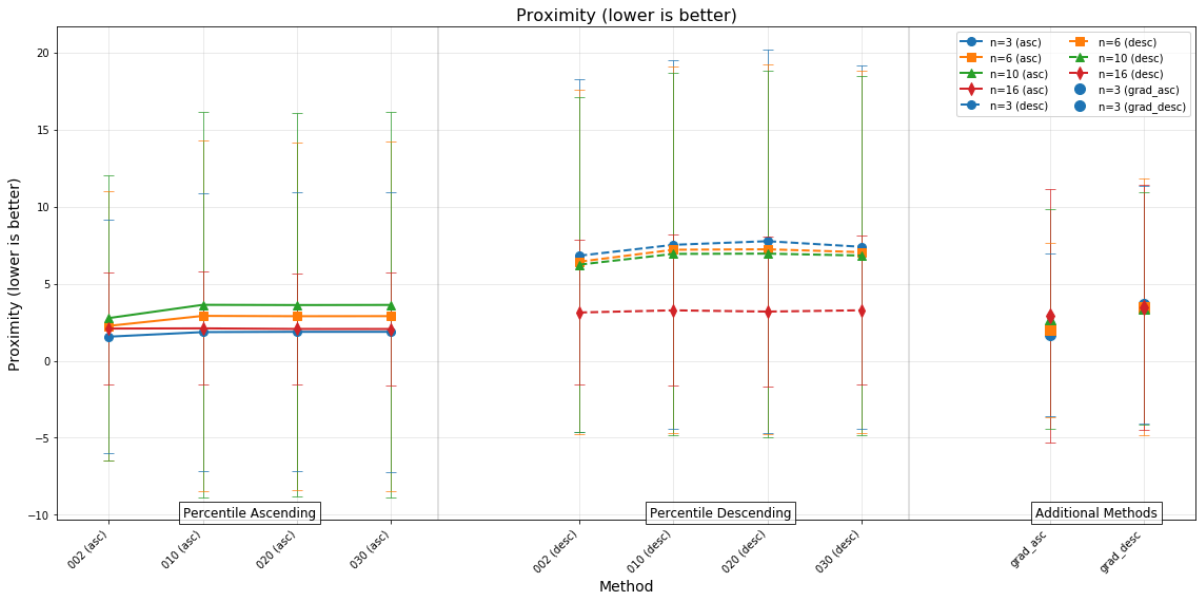


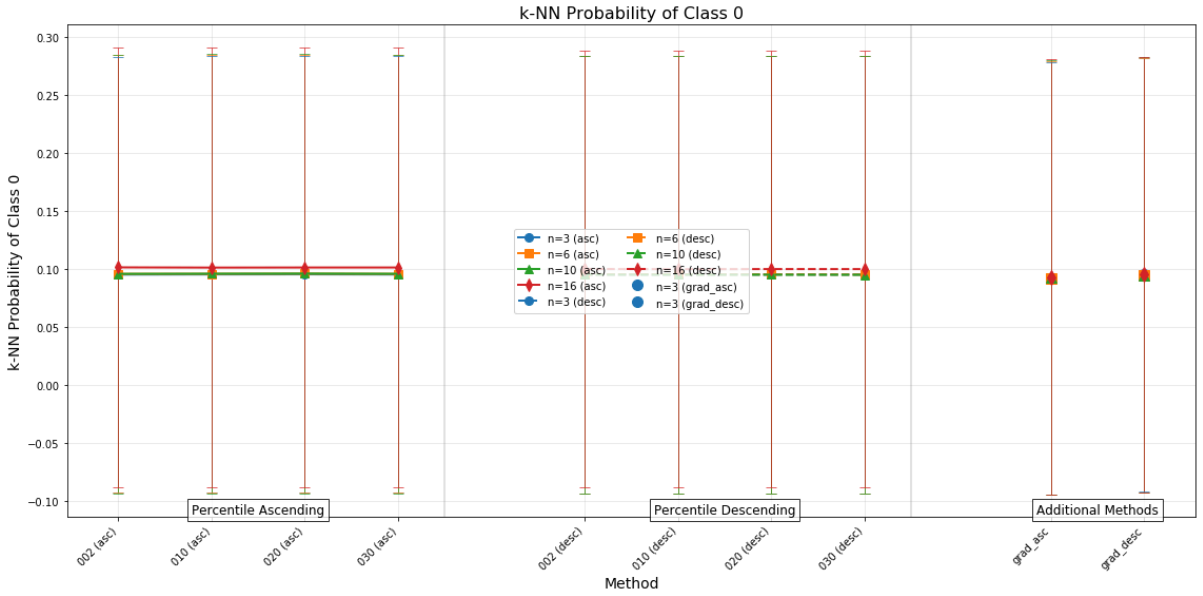
and for sparsity :

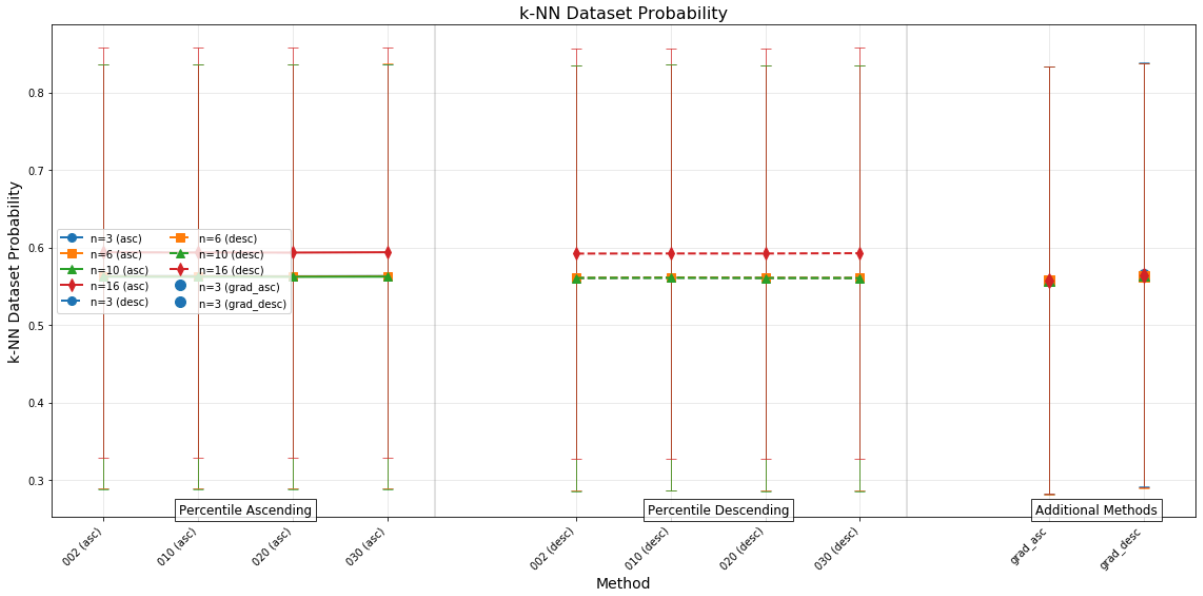



Sparsity Count (number of changed features, lower is better)
Sparsity Count (number of changed features, lower is better)
n=3 (asc)
n=6 (asc)
n=10 (asc)
n=16 (asc)
n=3 (desc)
n=6 (desc)
n=10 (desc)
n=16 (desc)
n=3 (grad_asc)
n=3 (grad_desc)
Percentile Ascending
Percentile Descending
Additional Methods
002 (asc)
010 (asc)
020 (asc)
030 (asc)
002 (desc)
010 (desc)
020 (desc)
030 (desc)
grad_asc
grad_desc
Method


Diversity (higher is better)
Diversity (higher is better)
n=3 (asc)
n=6 (asc)
n=10 (asc)
n=16 (asc)
n=3 (desc)
n=6 (desc)
n=10 (desc)
n=16 (desc)
n=3 (grad_asc)
n=3 (grad_desc)
Percentile Ascending
Percentile Descending
Additional Methods
002 (asc)
010 (asc)
020 (asc)
030 (asc)
002 (desc)
010 (desc)
020 (desc)
030 (desc)
grad_asc
grad_desc
Method

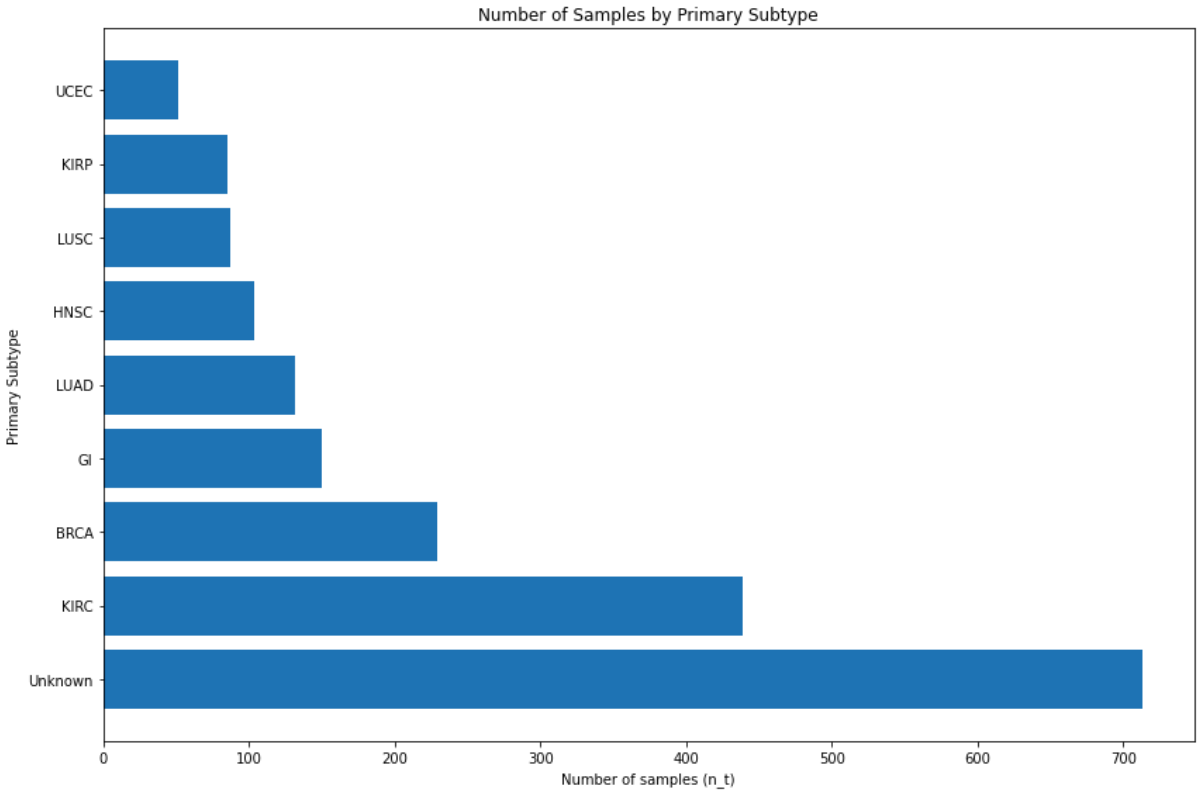

Proximity (lower is better)
Proximity (lower is better)
n=3 (asc)
n=6 (asc)
n=10 (asc)
n=16 (asc)
n=3 (desc)
n=6 (desc)
n=10 (desc)
n=16 (desc)
n=3 (grad_asc)
n=3 (grad_desc)
Percentile Ascending
Percentile Descending
Additional Methods
002 (asc)
010 (asc)
020 (asc)
030 (asc)
002 (desc)
010 (desc)
020 (desc)
030 (desc)
grad_asc
grad_desc
Method

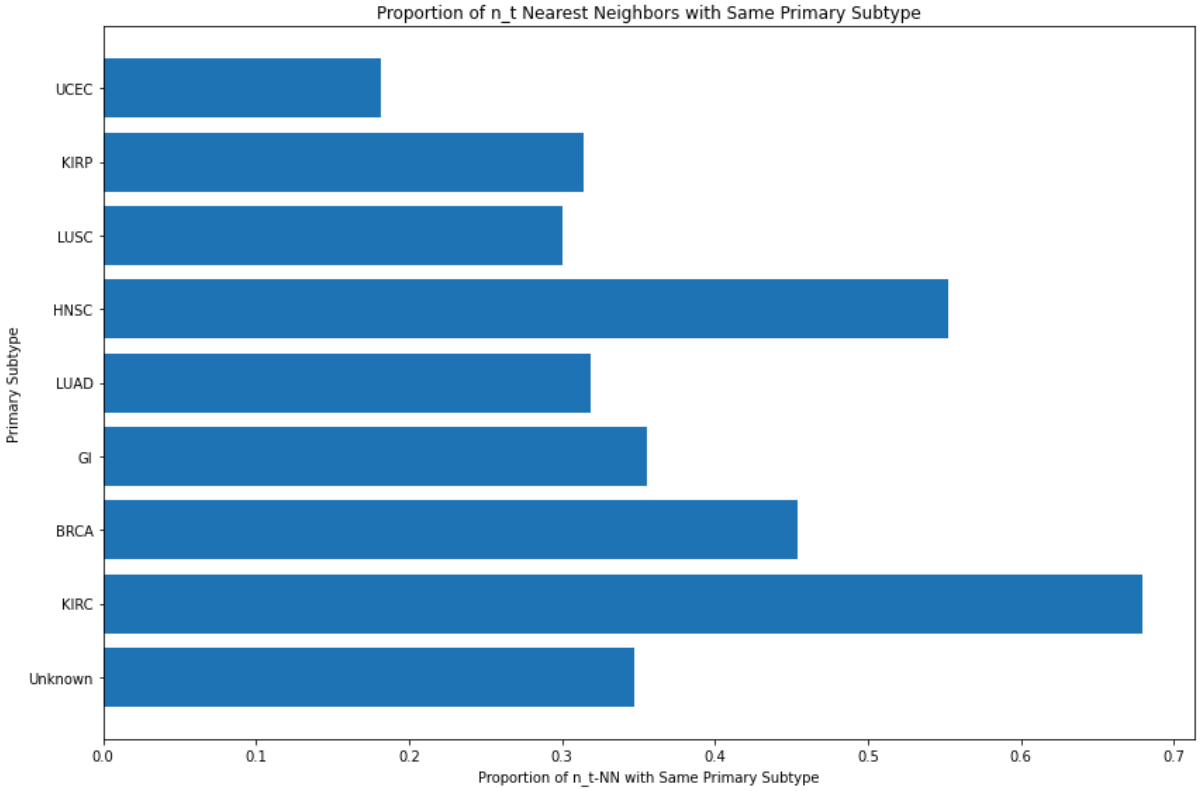




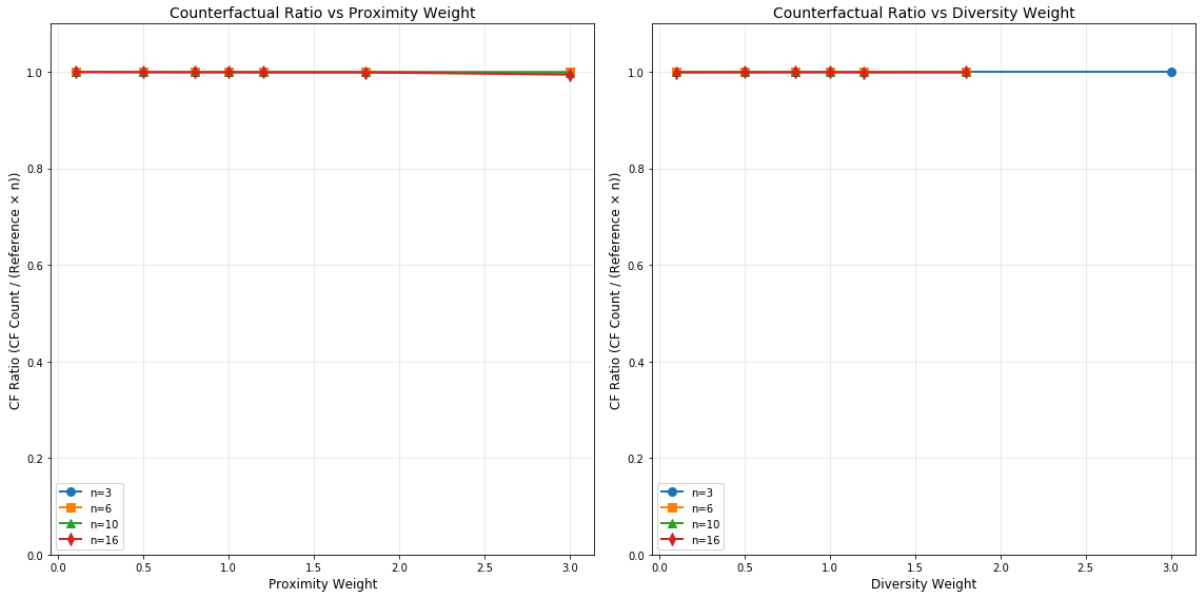
Brute force scoring function :
10-NN for plausibiliy
L1 distance for proximity
Classification value
Ces valuers sont chacunes normalisées.

Méthode : varier dans une range de +-10 pourcents du min et max de chaque feature classée 0, une, deux voire trois vecteurs à la fois, et y associer un score.
Exclure tous les points classée en dessous d'un seuil (supérieur à 0.3), ou à plus de 5 écarts types de la distance moyenne de 10-NN.
Pour chaque sample classe 1, on parcours le grille ainsi définie, échantillonnée en k valeurs.

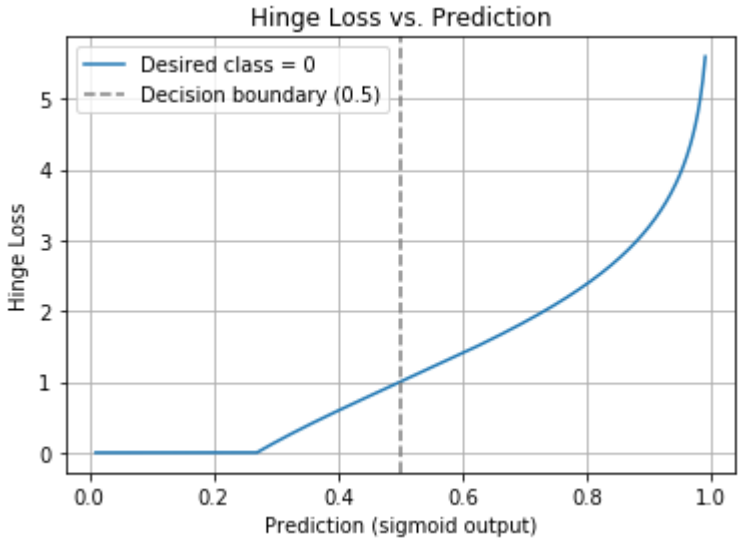

Number of Samples by Primary Subtype
Primary Subtype
UCEC
KIRP
LUSC
HNSC
LUAD
GI
BRCA
KIRC
Unknown
0
100
200
300
400
500
600
700
Number of samples (n_t)


Proportion of n_t Nearest Neighbors with Same Primary Subtype
Primary Subtype
UCEC
KIRP
LUSC
HNSC
LUAD
GI
BRCA
KIRC
Unknown
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Proportion of n_t-NN with Same Primary Subtype

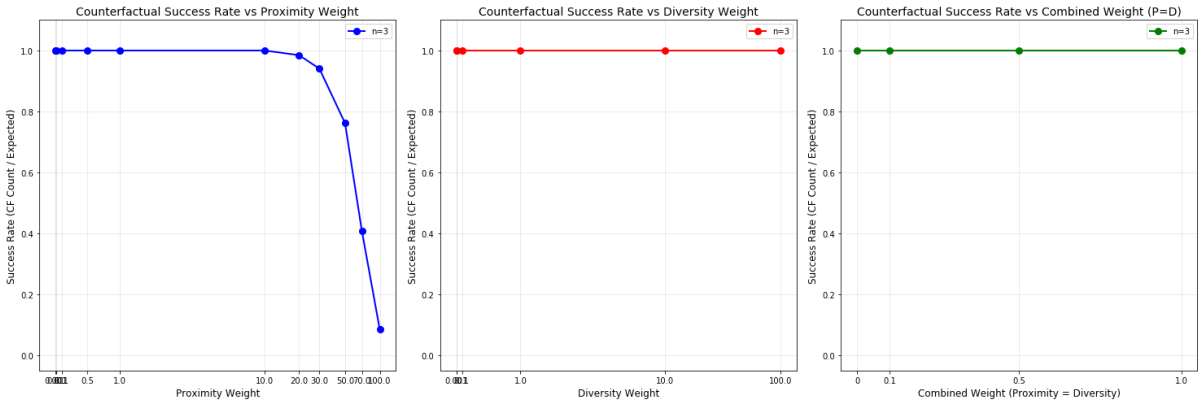
L’algorithme de génération de cf exclut les éléments invalides, mais la qualité de cf produits reste très haute : (toujours >0.999), pour n=16 cf (on a en effet 16<<20000).



La loss étant une hinge loss, elle n’encourage pas une validité supérieure une fois la valeur de classification x=1/(1+e)~0.269, ce qui est cohérent avec les résultats obtenus, et ce qui permet de coller à un objectif de distance à une frontière, ce qui se fait au détriment de la plausibilité, étant donné que la frontière semble fortement tirée du côté des points de classe cancer.



complexité bf : s*(np)^v
avec : s nombre de samples, n le nombre de features, p le nombre de valeurs testées par feature, v le nombre de features variant simultanément.

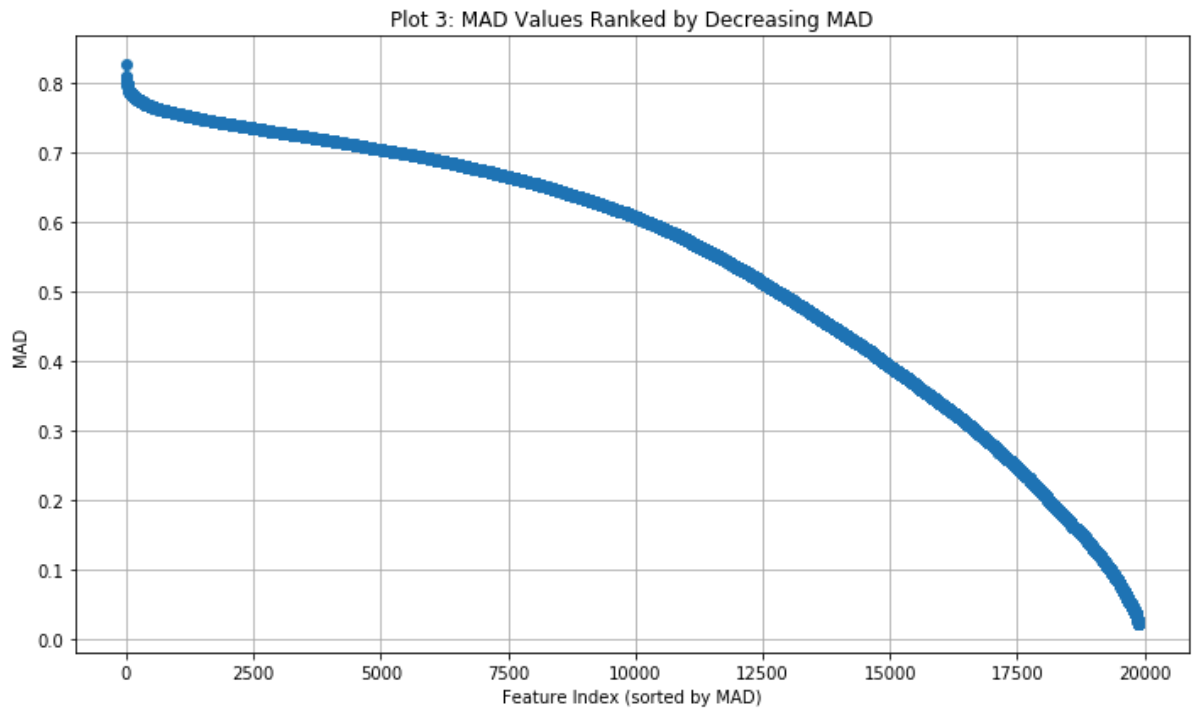


Variation du nombre de cfs valides en fonction des paramètres. Sans surprise, la proximité est le facteur limitant avec un seuil de dégradation à p=10.

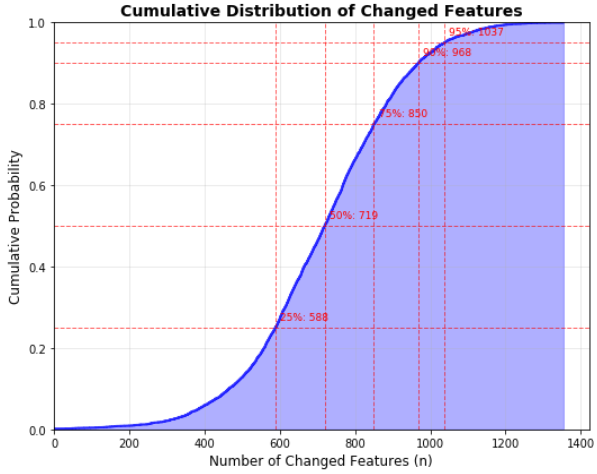
Brute force with n=1 results : 2 /1700 valid cfs when testing for 50 points per feature. Estimation for 2 features, test with 3 points : 5-10h / sample ⇒ min 20 days for 100 samples (closer to 40) one one GPU. Seems unrealistic to go for 3 features varying even with more gpus.

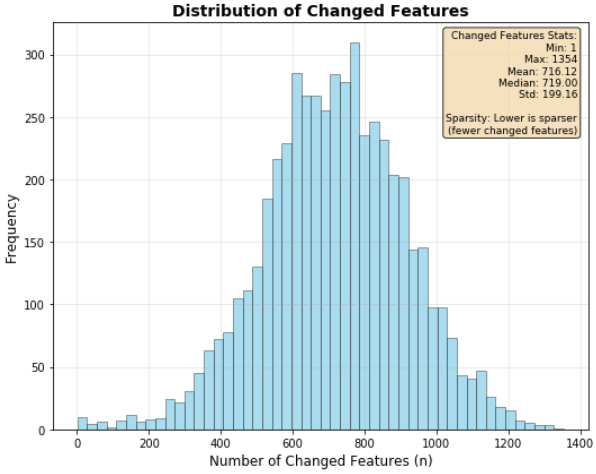
complexité théorique bf : s*(np)^v
avec : s nombre de samples, n le nombre de features, p le nombre de valeurs testées par feature, v le nombre de features variant simultanément.

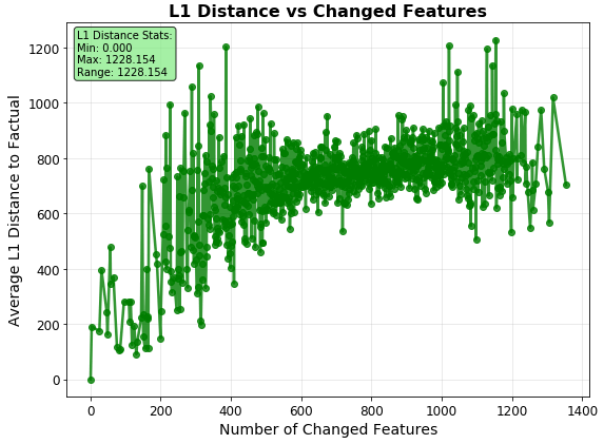
2 options : sampler les vecteurs et garder les composante de variance supérieure

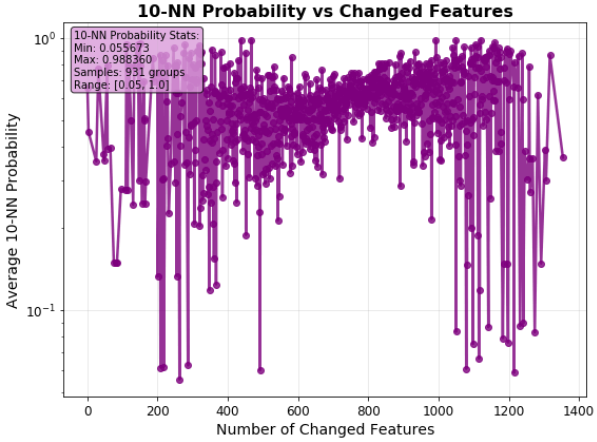


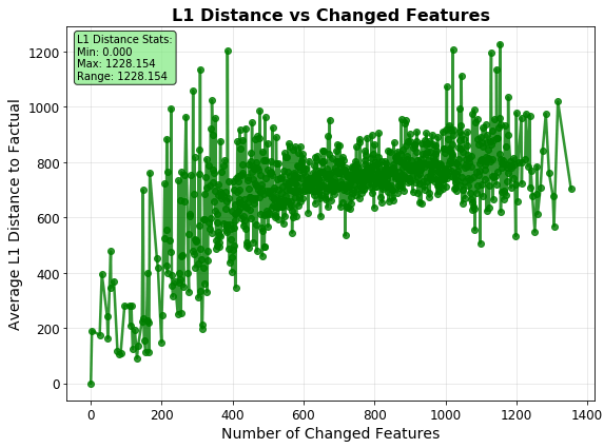
And we get the cdf for sparsity for 3 counterfactuals :

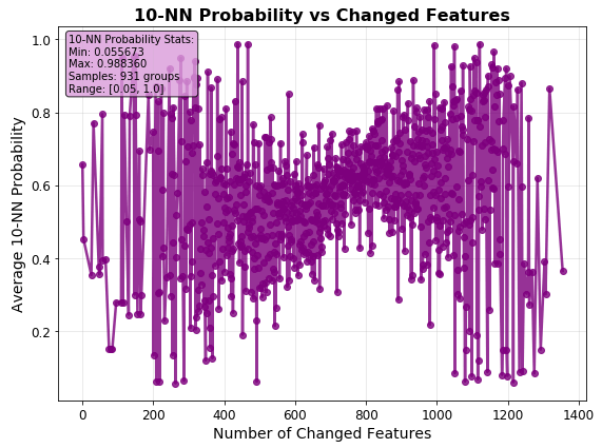


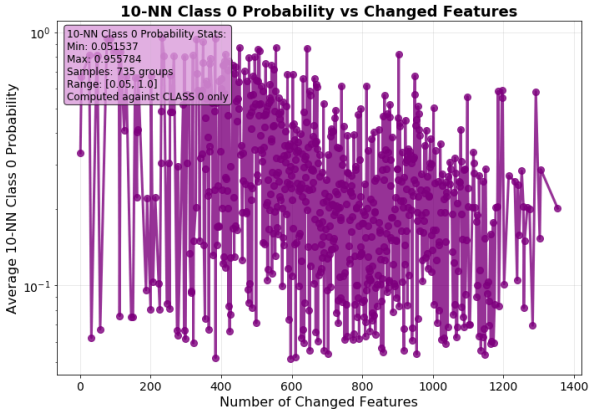


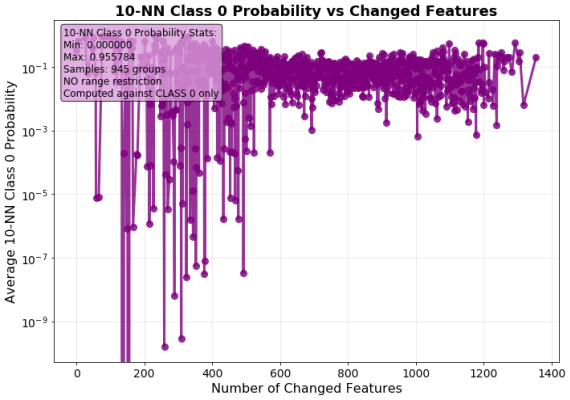




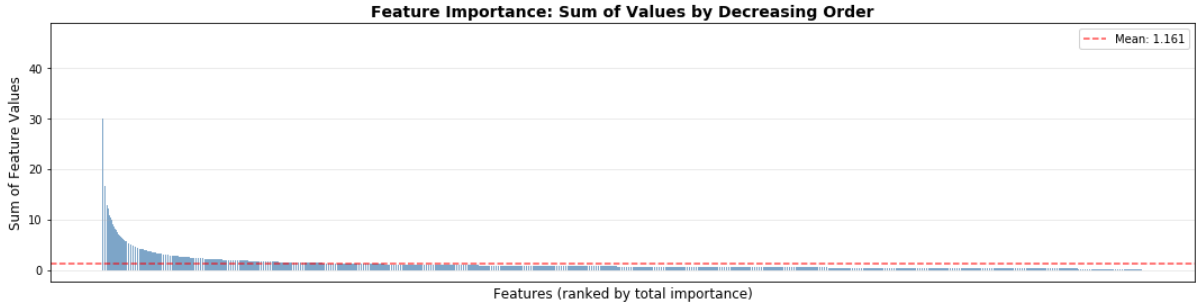


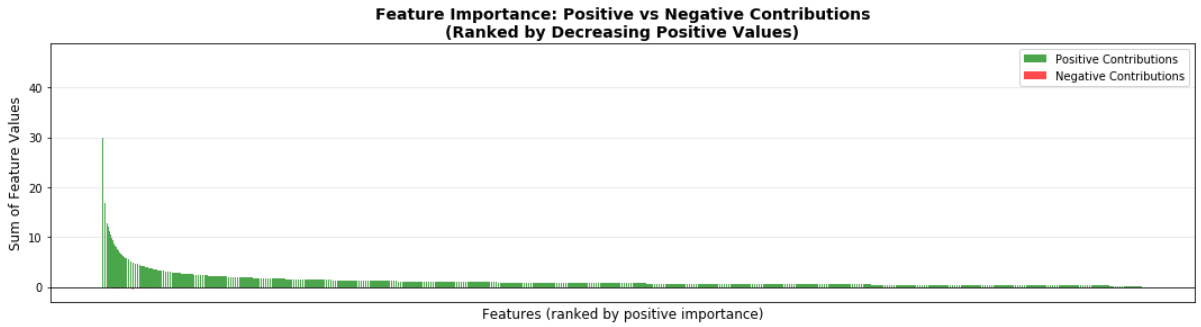






Feature Importance Analysis :
computed as 30*(M(x')-M(x)) + 10*abs(x'-x))/std(L1) +
10*abs(10NN(x')-avg(10NN))/std(10NN)

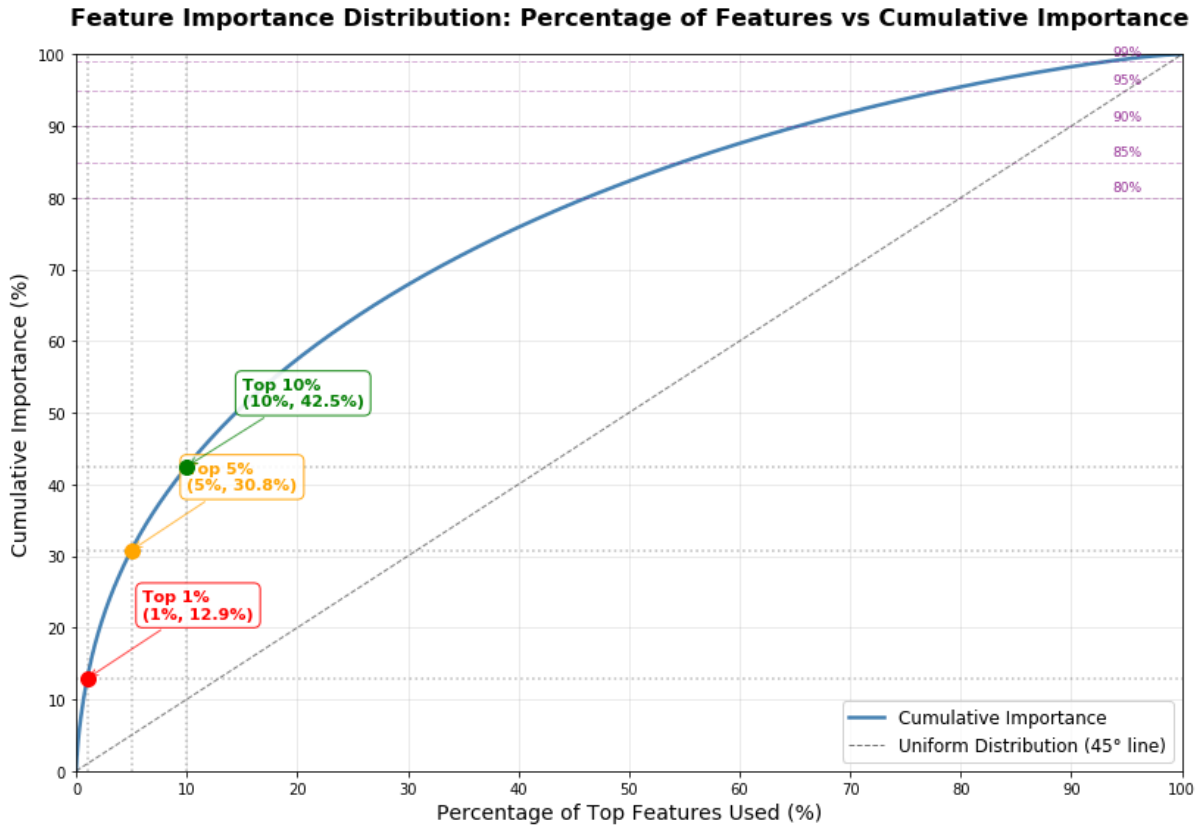




Top 10 features by total importance:
feature_18042_diff 46.431678
feature_6049_diff 40.509097
feature_15302_diff 39.305724
feature_3070_diff 34.855112
feature_12440_diff 34.074335
feature_19441_diff 33.442429
feature_6251_diff 31.125915
feature_5459_diff 30.375266
feature_15805_diff 29.879331
feature_16831_diff 29.137779
dtype: float64

Top 10 features by positive contributions:

feature positive_sum



feature_18042_diff 46.463460
feature_6049_diff 40.532533
feature_15302_diff 39.336990
feature_3070_diff 34.897965
feature_12440_diff 34.172294
feature_19441_diff 33.505872
feature_6251_diff 31.179756
feature_5459_diff 30.498958
feature_15805_diff 29.946055
feature_16831_diff 29.150749

Features with highest negative contributions:

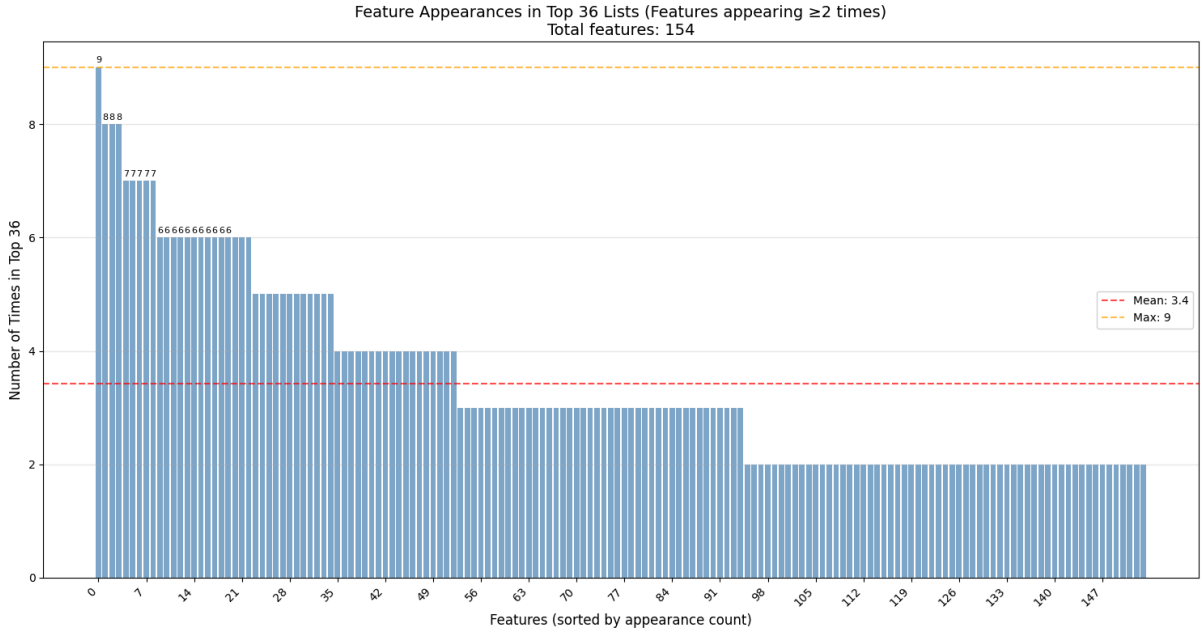
feature negative_sum
feature_12171_diff -0.621873
feature_5579_diff -0.619893
feature_16698_diff -0.599867
feature_7500_diff -0.580269
feature_4051_diff -0.551151
feature_17004_diff -0.539751
feature_17119_diff -0.523621
feature_19208_diff -0.522294
feature_15184_diff -0.518933
feature_3406_diff -0.513319

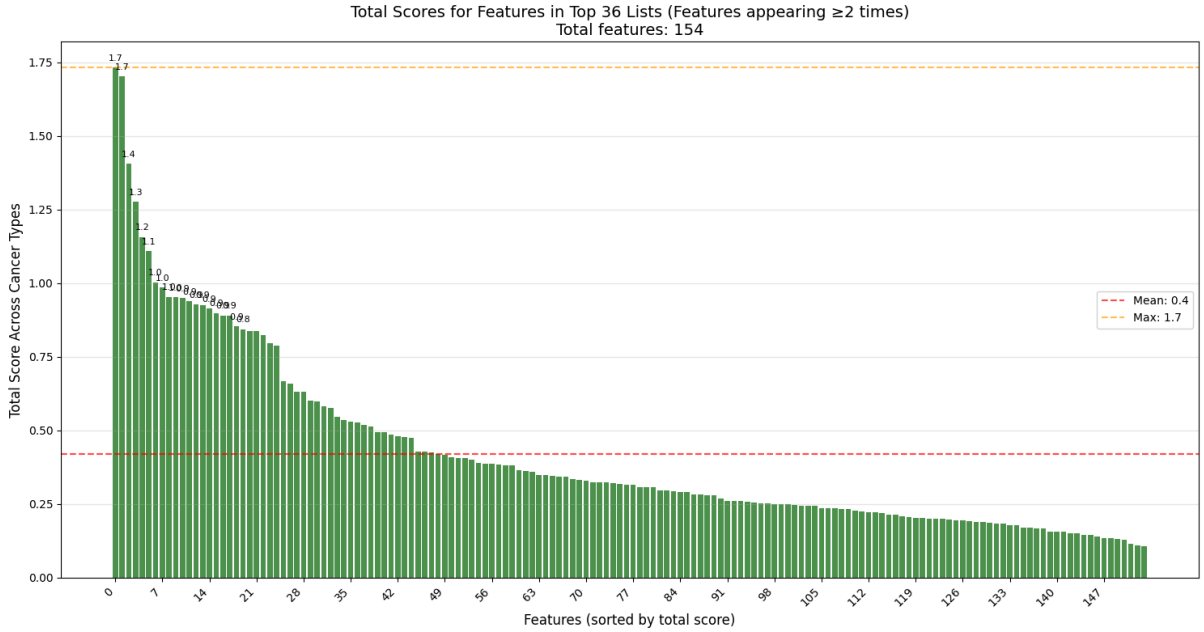
Overall statistics:
Total positive contributions: 24350.160

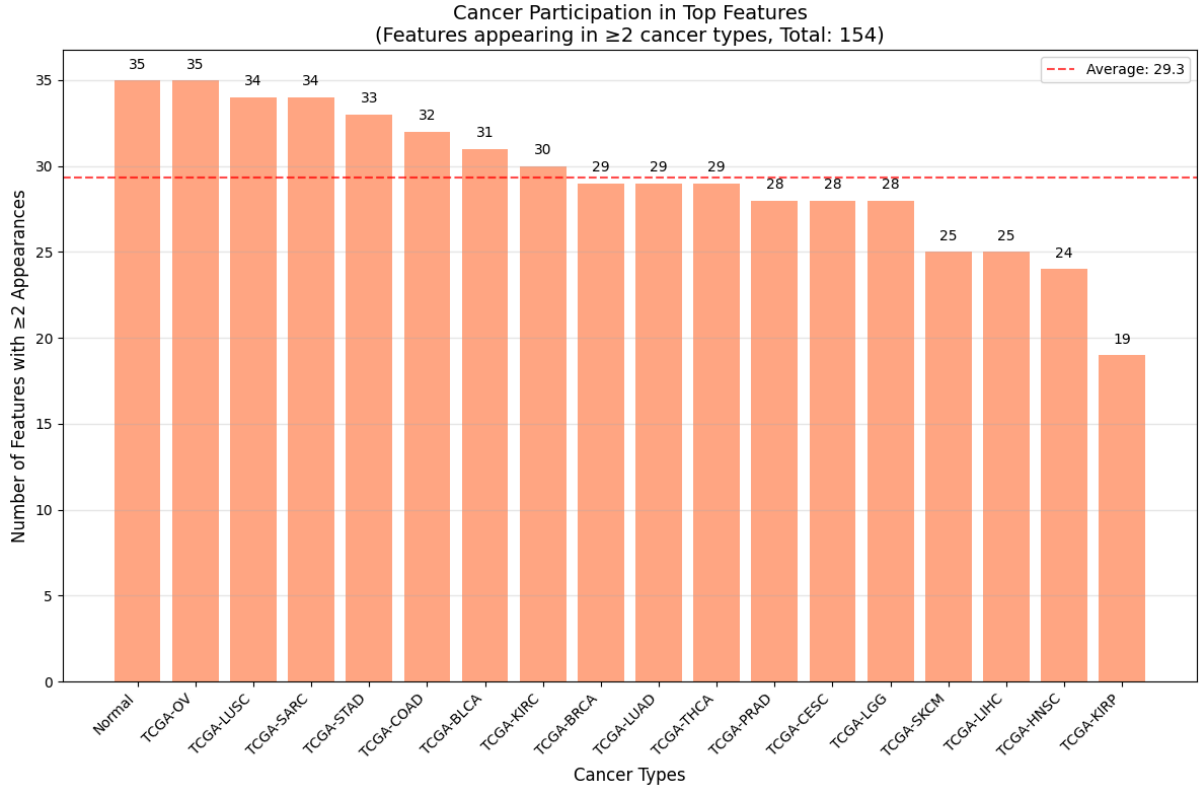
Total negative contributions: -1266.657
Net importance: 23083.503
Number of features with positive contributions: 19887
Number of features with negative contributions: 19887

Time to Run BF algo :
1 feature, 25 points, 1700 samples : 1 week
2 features, 6*4 points, 100 samples, top 2 percent of features explaining 20 percent of importance : 2 weeks

Launched also a second run of the 1 feature with the best “counterfactuals” of the previous 1 feature bf run to emulate a 2 feature bf heuristic ; jumped from 4 to 8 valid counterfactuals.

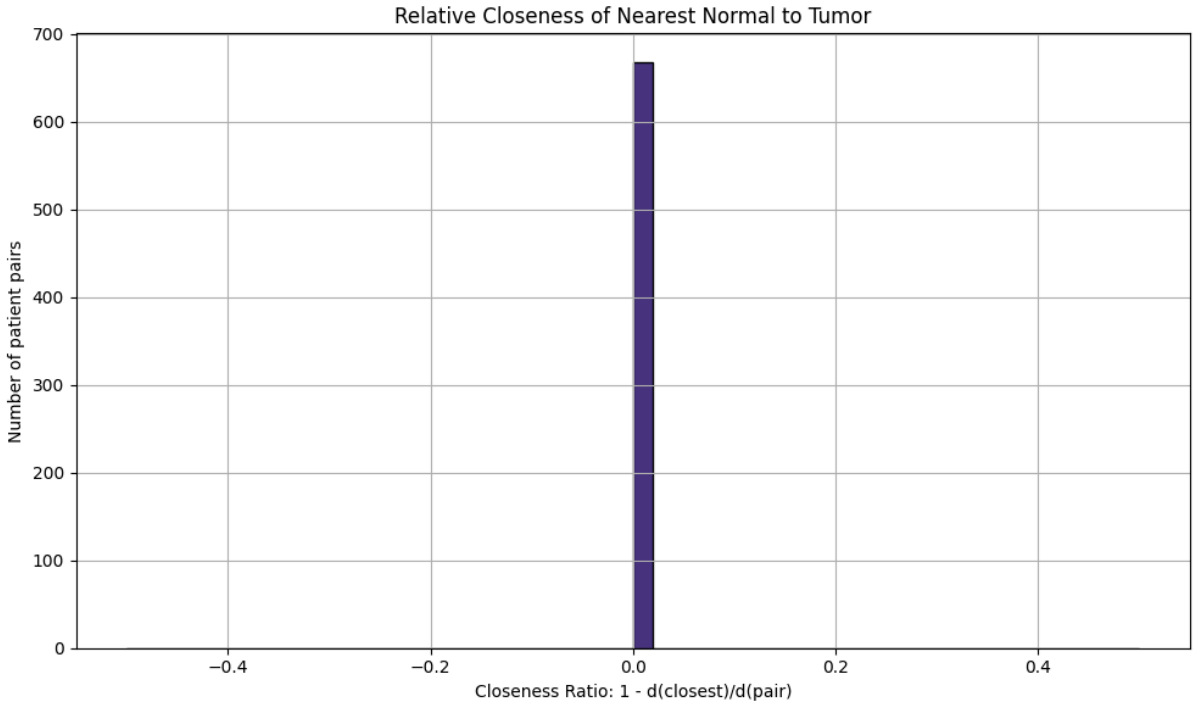


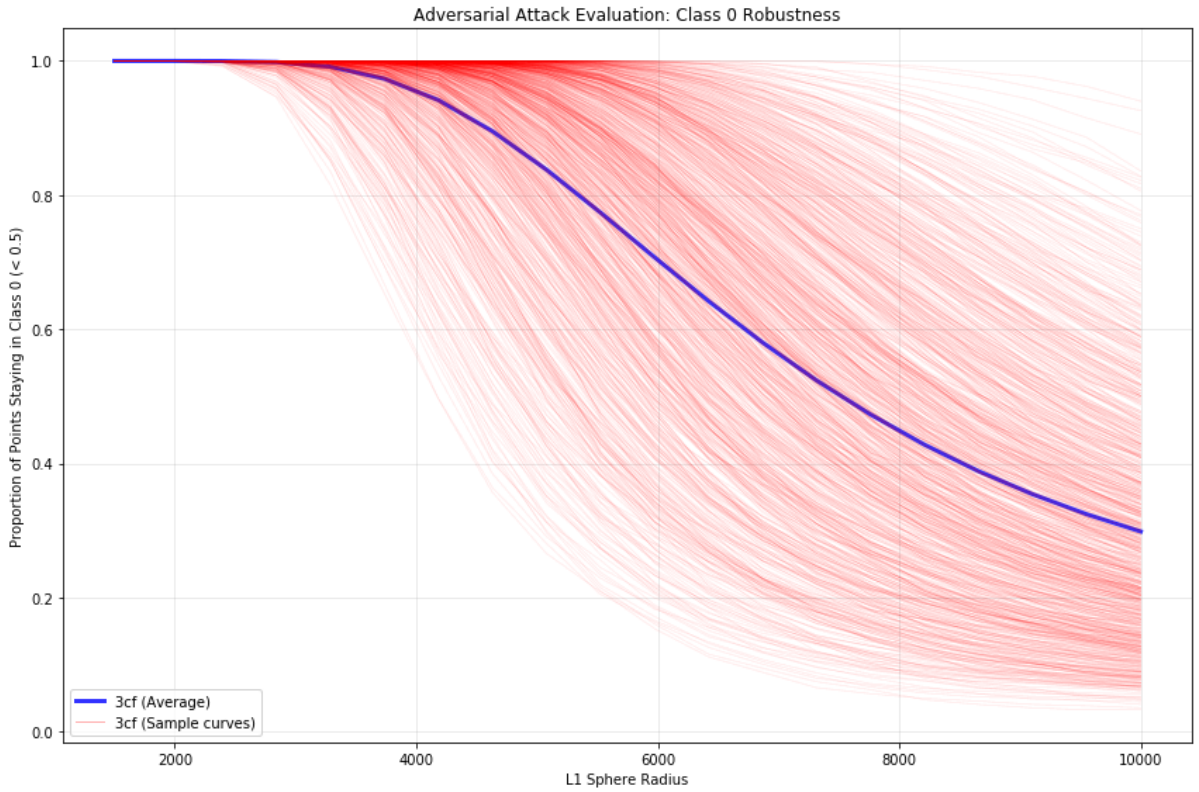


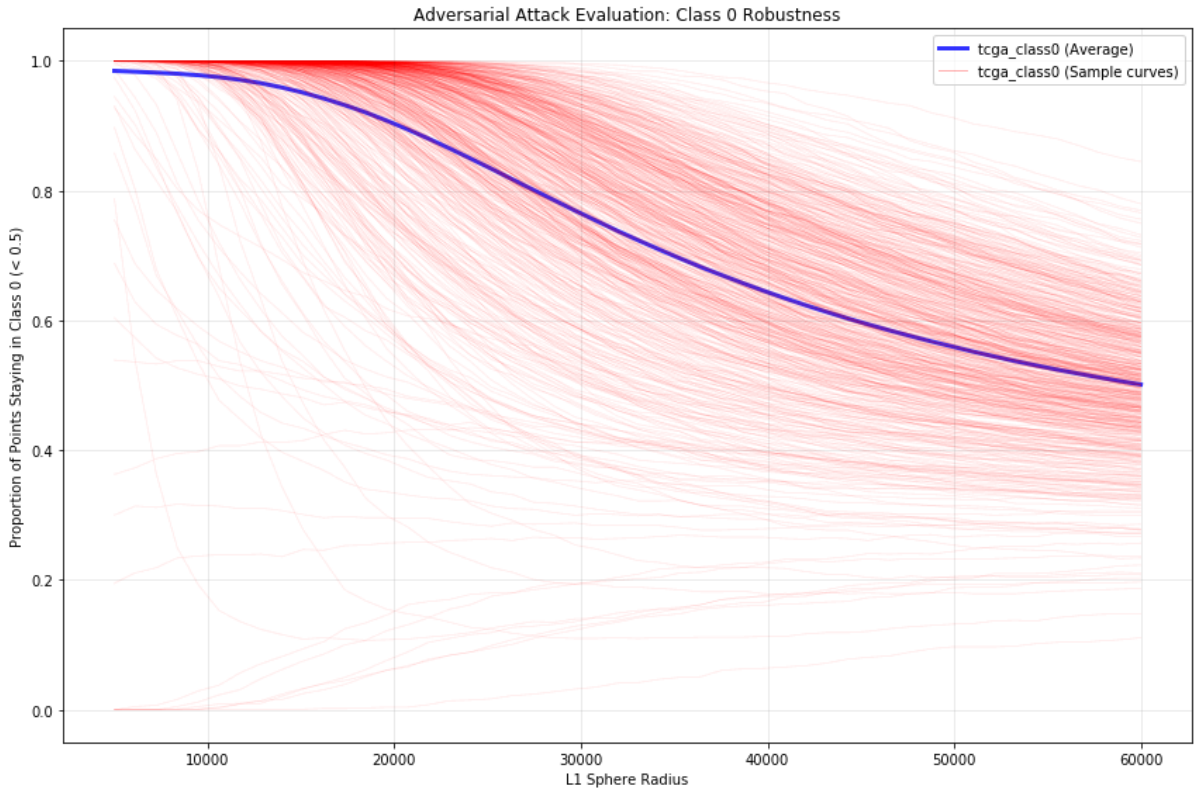


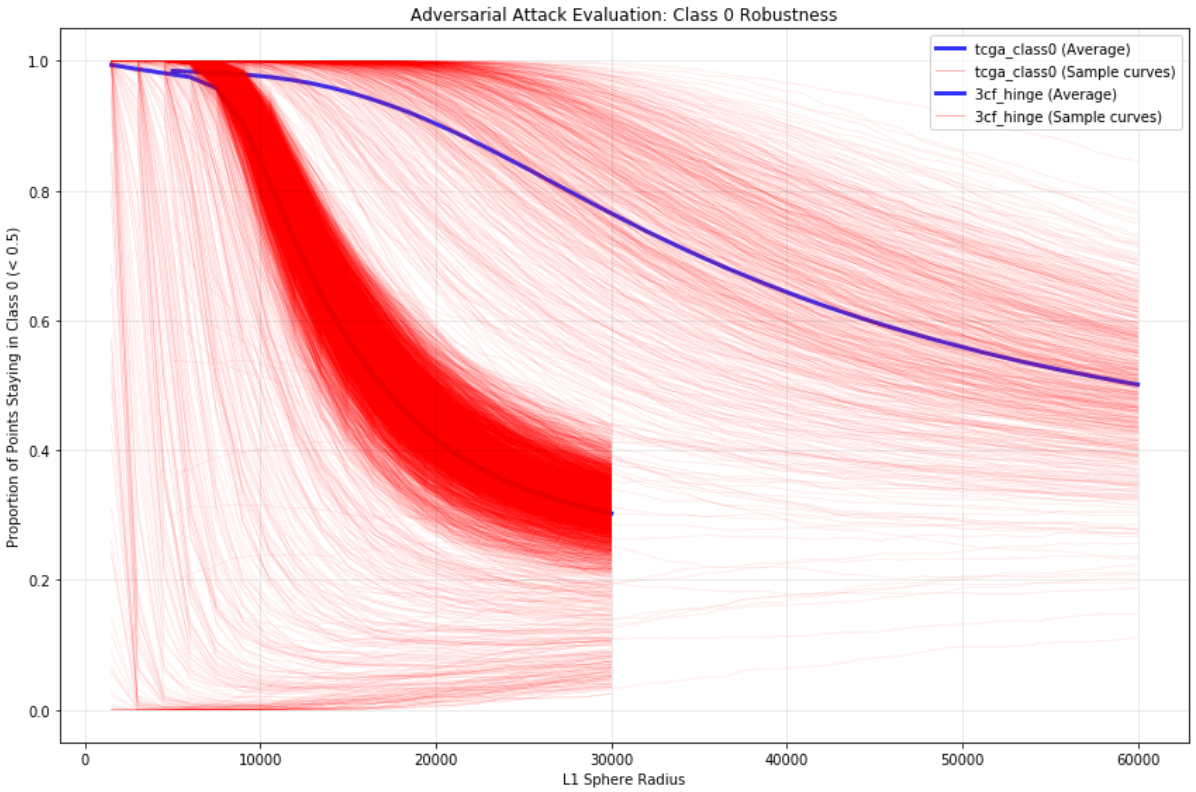
Computed top 36 features for each of 18 cancers. Plotted most frequent features among the top 36 as number of times represented and then cumulated importance.
The idea is to differentiate if certain features are specific to certain cancers. When taking top 36 features for each cancer and looking for overlap of features that appear more that twice, we discern 154 features that behave like this.
Reversing the analysis, these 154 features represent systematically between 19 and 35 of the top 36 features (average 29,3).
The idea could now be to see how the top features for all cancers and the proportion of

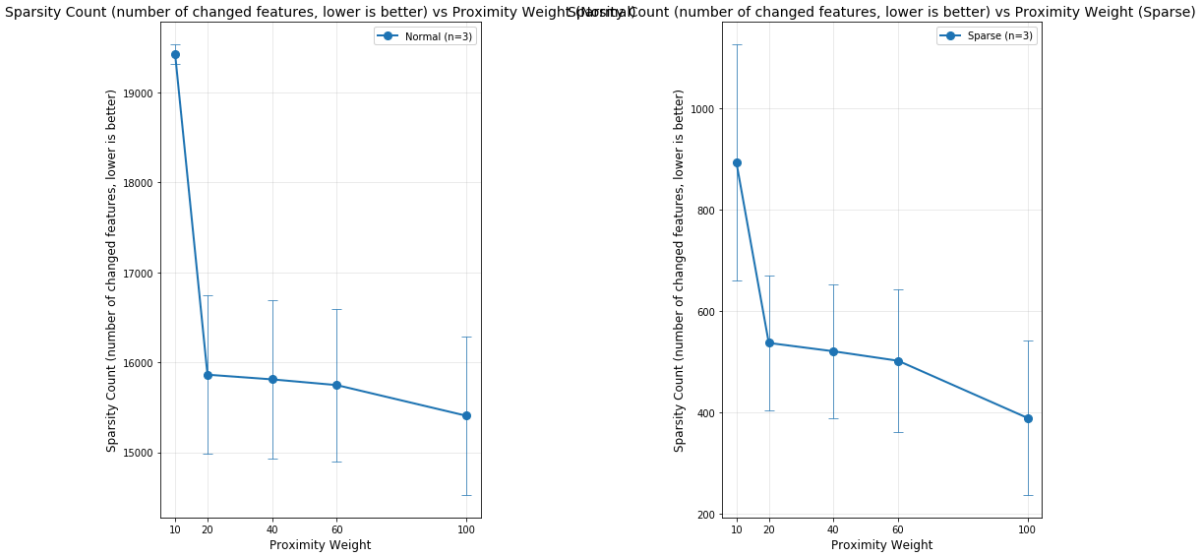
these feature in a given cancer intercat, also given the limit we set, 2, which could be increase for stronger results.

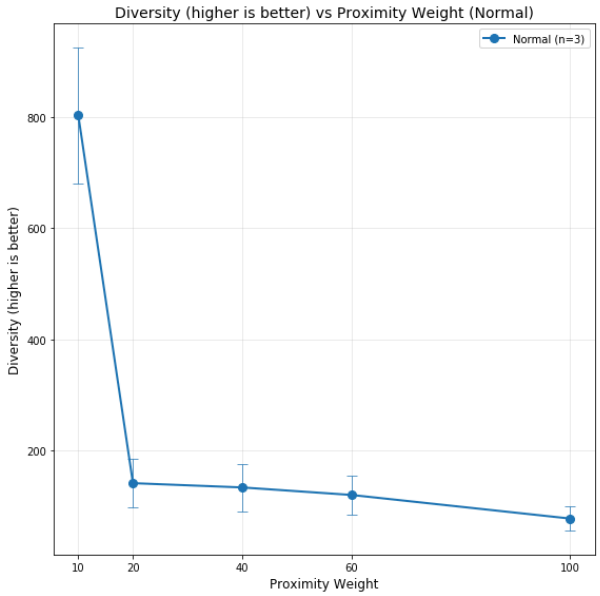


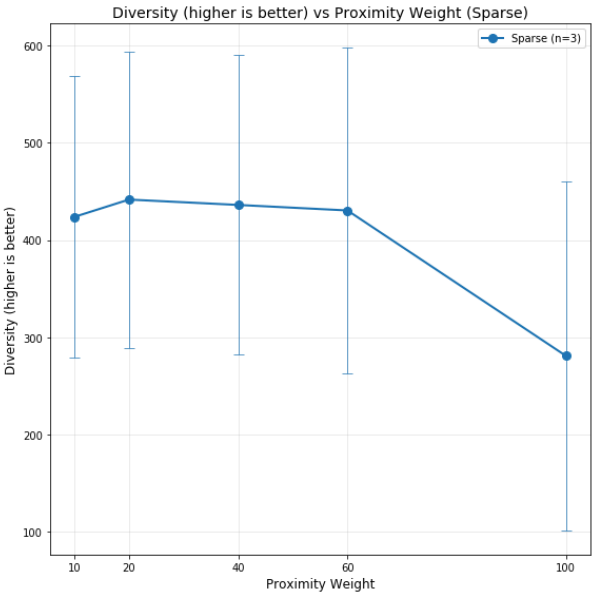

Adversarial Attack Evaluation: Class 0 Robustness
Proportion of Points Staying in Class 0 (< 0.5)
L1 Sphere Radius
3cf (Average)
3cf (Sample curves)
1.0
0.8
0.6
0.4
0.2
0.0
2000
4000
6000
8000
10000

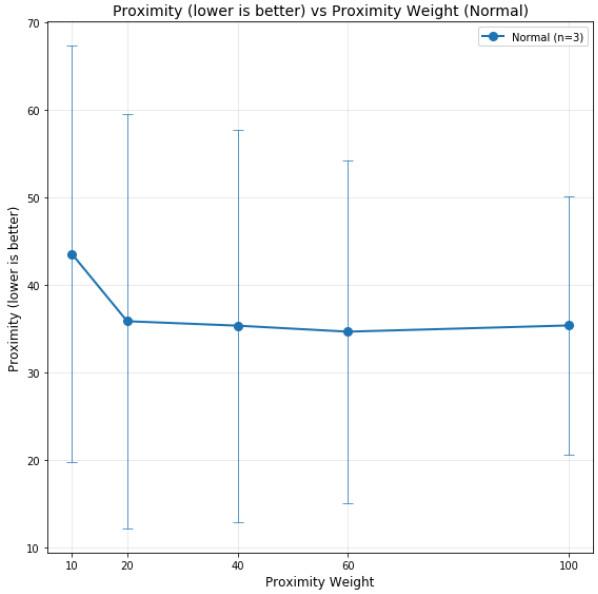

Adversarial Attack Evaluation: Class 0 Robustness
Proportion of Points Staying in Class 0 (< 0.5)
L1 Sphere Radius
tcga_class0 (Average)
tcga_class0 (Sample curves)
1.0
0.8
0.6
0.4
0.2
0.0
10000
20000
30000
40000
50000
60000

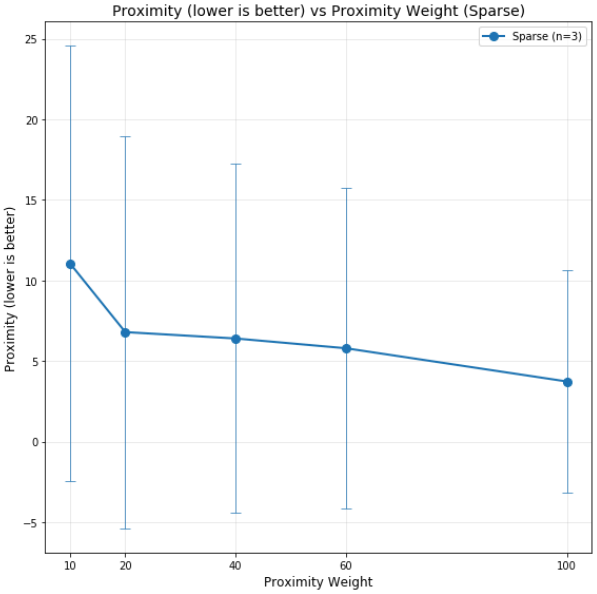

Adversarial Attack Evaluation: Class 0 Robustness
tcga_class0 (Average)
tcga_class0 (Sample curves)
3cf_hinge (Average)
3cf_hinge (Sample curves)
Proportion of Points Staying in Class 0 (< 0.5)
L1 Sphere Radius
0.0
0.2
0.4
0.6
0.8
1.0
0
10000
20000
30000
40000
50000
60000

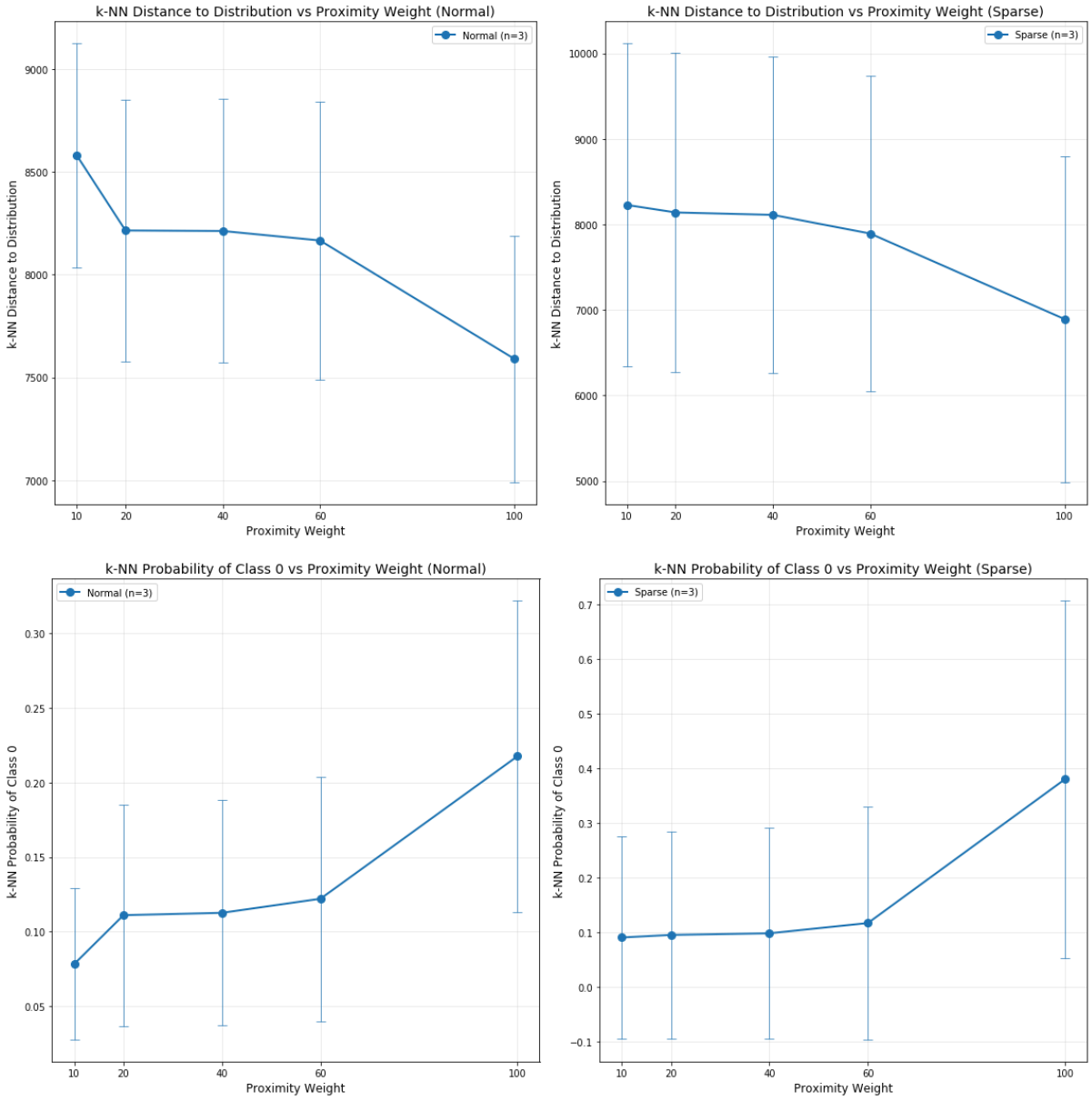

Sparsity Count (number of changed features, lower is better) vs Proximity Weight (Normal)
Normal (n=3)
Sparsity Count (number of changed features, lower is better)
Proximity Weight
15000
16000
17000
18000
19000
10
20
40
60
100
Sparsity Count (number of changed features, lower is better) vs Proximity Weight (Sparse)
Sparse (n=3)
Sparsity Count (number of changed features, lower is better)
Proximity Weight
200
400
600
800
1000
10
20
40
60
100


Diversity (higher is better) vs Proximity Weight (Normal)
Normal (n=3)
Diversity (higher is better)
Proximity Weight
800
600
400
200
10
20
40
60
100


Diversity (higher is better) vs Proximity Weight (Sparse)
Sparse (n=3)
Diversity (higher is better)
Proximity Weight
600
500
400
300
200
100
10
20
40
60
100


Proximity (lower is better) vs Proximity Weight (Normal)
Normal (n=3)
Proximity (lower is better)
Proximity Weight
70
60
50
40
30
20
10
10
20
40
60
100


Proximity (lower is better) vs Proximity Weight (Sparse)
Sparse (n=3)
Proximity (lower is better)
Proximity Weight
25
20
15
10
5
0
−5
10
20
40
60
100



3cf_hinge:
  validity: min=0.0000, avg=0.9974, max=1.0000, std=0.0513
  proximity: min=21.2763, avg=37.4756, max=251.0544, std=22.3904
  sparsity: min=0.0493, avg=0.1994, max=0.3419, std=0.0430
  sparsity_count: min=13088.0000, avg=15921.4043, max=18907.0000, std=855.2889
  diversity: min=74.1638, avg=212.3355, max=515.3057, std=67.1131
  sparse_diversity: min=0.0003, avg=0.0015, max=0.0029, std=0.0004
  avg_10nn_distance: min=6606.7773, avg=8309.6611, max=10144.6660, std=628.9130
  avg_10nn_dataset_probability: min=0.3964, avg=0.8249, max=0.9998, std=0.1369
  avg_10nn_class0_probability: min=0.0040, avg=0.1040, max=0.5273, std=0.0702

sparse_3cf_hinge:
  validity: min=0.0000, avg=0.9974, max=1.0000, std=0.0513
  proximity: min=0.0000, avg=8.4788, max=130.4135, std=11.4713
  sparsity: min=0.9428, avg=0.9763, max=1.0000, std=0.0064

sparsity_count: min=0.0000, avg=470.6148, max=1137.0000, std=127.3480
diversity: min=0.0000, avg=353.7324, max=1971.1400, std=148.7891
sparse_diversity: min=0.4582, avg=0.6404, max=0.8520, std=0.0509
avg_10nn_distance: min=3746.1575, avg=8150.0156, max=22707.7754, std=1879.9458
avg_10nn_dataset_probability: min=0.0000, avg=0.5553, max=0.9997, std=0.2746
avg_10nn_class0_probability: min=0.0000, avg=0.0965, max=0.9989, std=0.1924