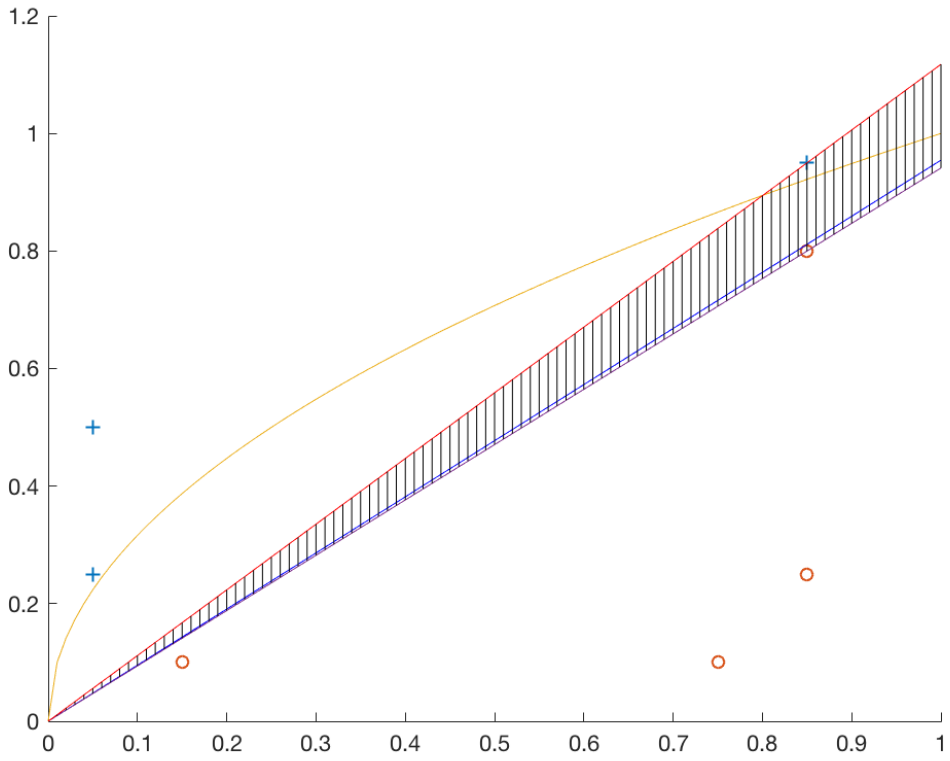
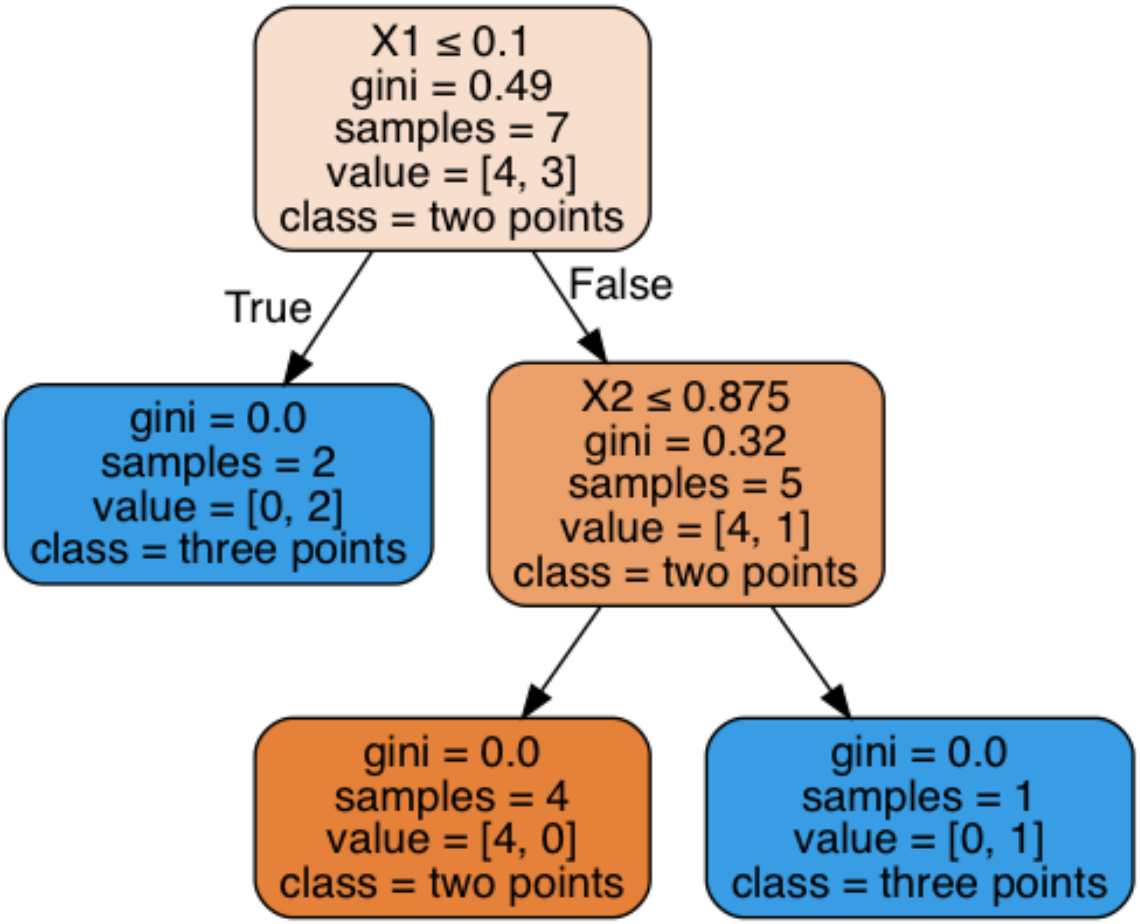
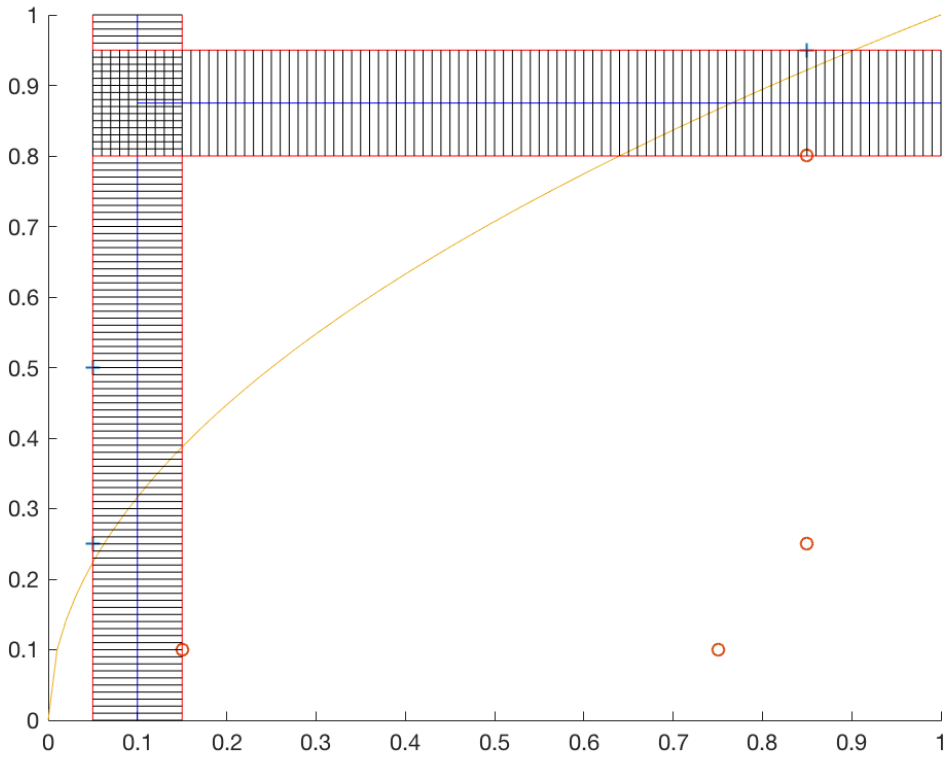
1 Classifiers for Basketball Courts

(a) It takes 7 updates to converge and the error of the classifier is 0. The weight vector we got from running perception algorithm is [-1.05, 1.1] (blue line). The lines in the shaded area (not including the two red lines) give the same error. They have the weight vector [w1, w2] where w2 = 1 and w1 ranges in (0.8/0.85, 0.9/0.85).



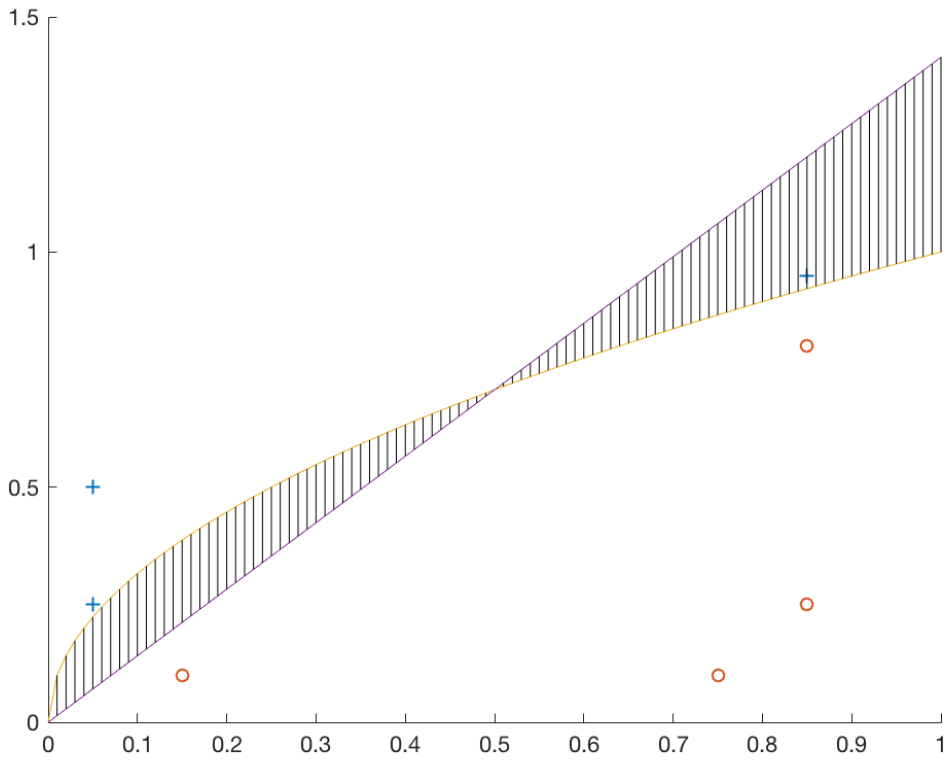
(b) We use sklearn library to build decision tree and use graphviz to visualize decision tree. In the graph below, samples means number of samples at that node and value represents vector of samples for each class. And this decision tree gives error 0 (accuracy 1).


X1 ≤ 0.1
gini = 0.49
samples = 7
value = [4, 3]
class = two points
True
False
gini = 0.0
samples = 2
value = [0, 2]
class = three points
X2 ≤ 0.875
gini = 0.32
samples = 5
value = [4, 1]
class = two points
gini = 0.0
samples = 4
value = [4, 0]
class = two points
gini = 0.0
samples = 1
value = [0, 1]
class = three points

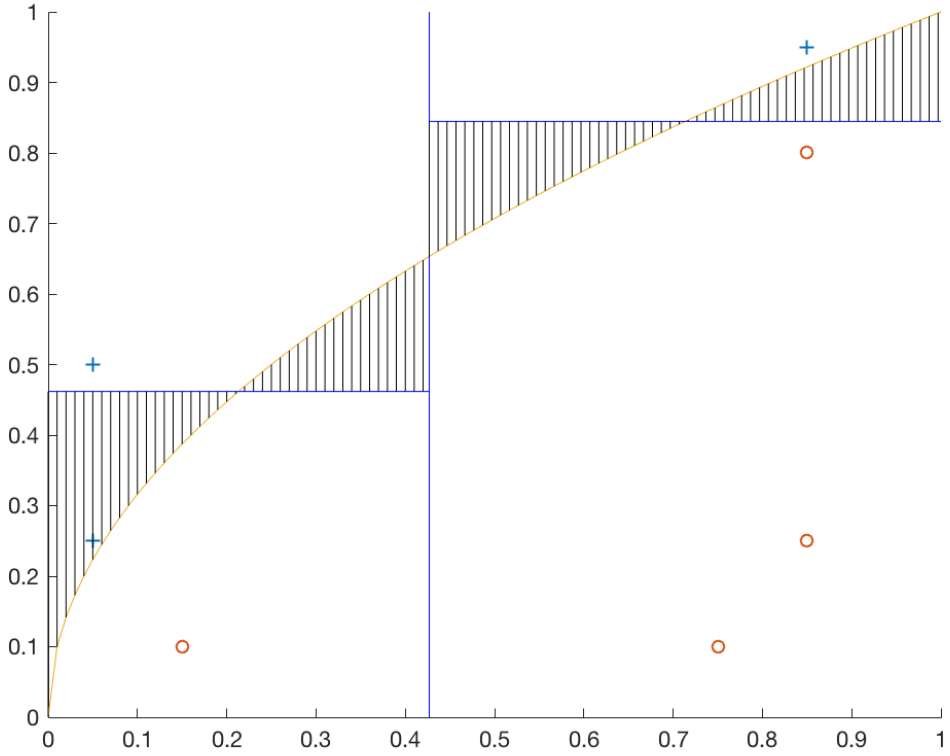


The decision boundary driven from the decision tree above are of blue color. Other solutions with the same accuracy fall in the shaded area (not including the four red lines). The first splitting criteria is x1⩽a, where a ranges in (0.05, 0.15). The second splitting criteria is x2⩽b, where b ranges in (0.8, 0.95).

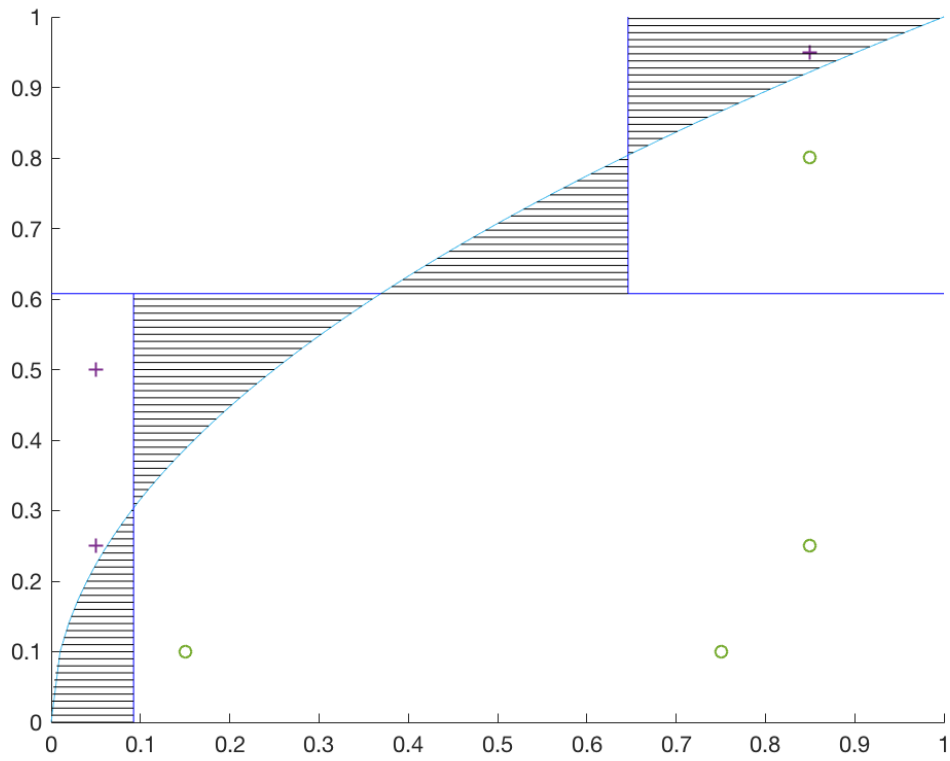
(c) We compute the k where $(\int_0^1 |\sqrt{x} - kx|\, dx - \frac{1}{2}(1 - \frac{1}{k})(k-1))' = 0$ (minimize the shaded area between $y = \sqrt{x}$ and $y = kx$) and k approximately equals to 1.41421356237310. So, the optimal weight vector is [-1.41421356237310 1]. And the true risk that this solution gives is 0.158291244717639.



(d) **If we split on x1 first**, we compute the (s1, s2, s2) that minimize $\int_0^{s_1}|\sqrt{x_1}-s_2|dx_1+\int_{s1}^{1}|\sqrt{x_1}-s_3|dx_1$ (the shaded area in the graph below) and (s1, s2, s3) approximately equals to (0.426777. 0.46194, 0.844623). And the true risk that this solution gives is approximately 0.103585.



If we split on x2 first, we compute the (s1, s2, s2) that minimize $\int_0^{s_1}|{x_2}^2 - s_2|dx_2 + \int_{s1}^1|{x_2}^2 - s_3|dx_2$ (the shaded area in the graph below) and (s1, s2, s3) approximately equals to (0.607625. 0.0923021, 0.646115). And the true risk that this solution gives is approximately 0.117962.



The tree which split on x1 first has the lower true risk, but this solution is not among the solutions that achieved the minimum empirical loss in part(b).

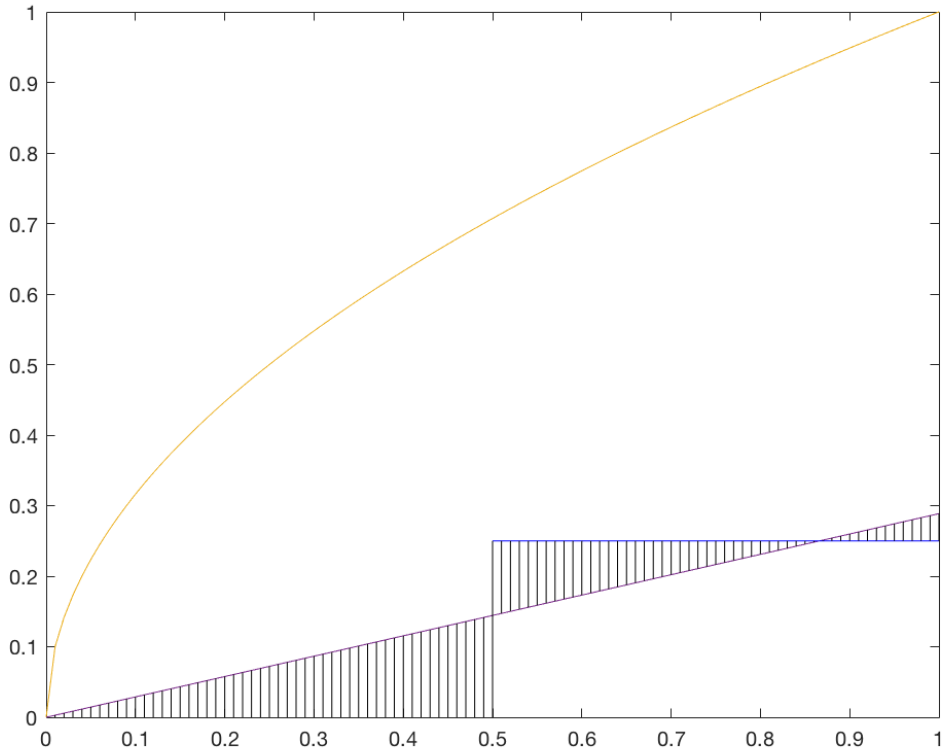
(e) The transformation is applying square to x_2 (changing it to ${x_2}^2$) or applying square to x_1 (changing it to $\sqrt{x_1}$) and the optimal weight vector is [-1, 1]. The classifier is: -1*X1 + 1*X2 = 0 and its error is zero.

(f) No.

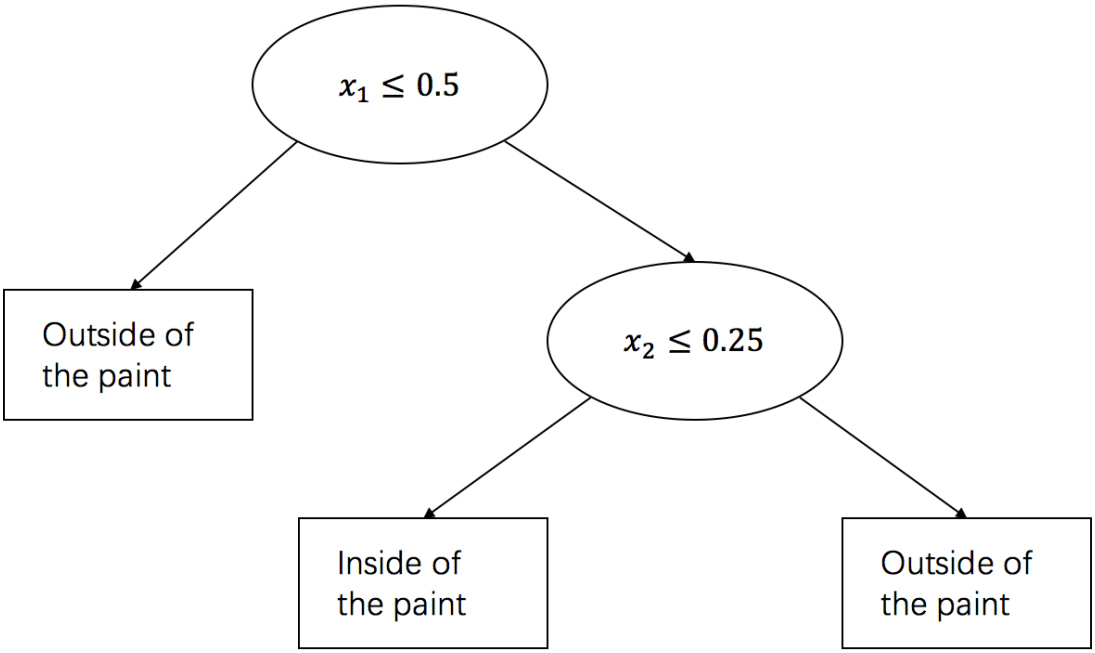
(g)

(h) For (h) and (i), a label of Y=1 means a point is outside of the paint while a label of Y=-1 means a point is inside of the paint. We compute the k where ($\int_0^{0.5} kxdx +$

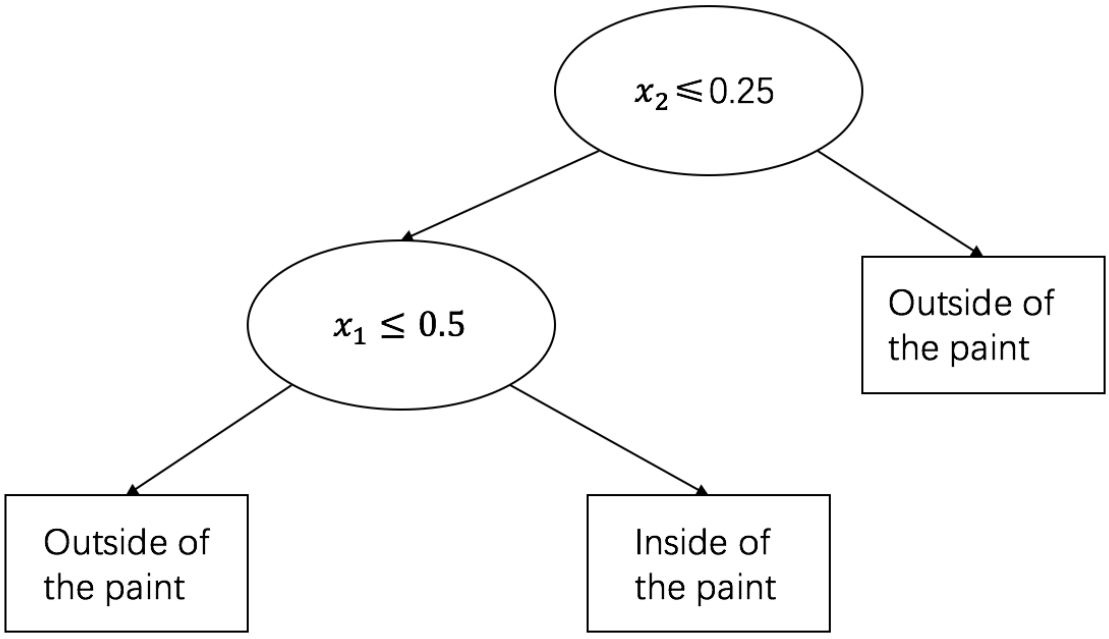
$\int_{0.5}^{1}|kx - 0.25|\, dx)' = 0$, and k approximately equals to 0.288675134594813. So the optimal weight vector is [-0.288675134594813 1]. And the true risk that this solution gives is 0.0580127018922193.



(i) **If we spit on x1 first**, the optimal depth 2 decision tree looks like below:



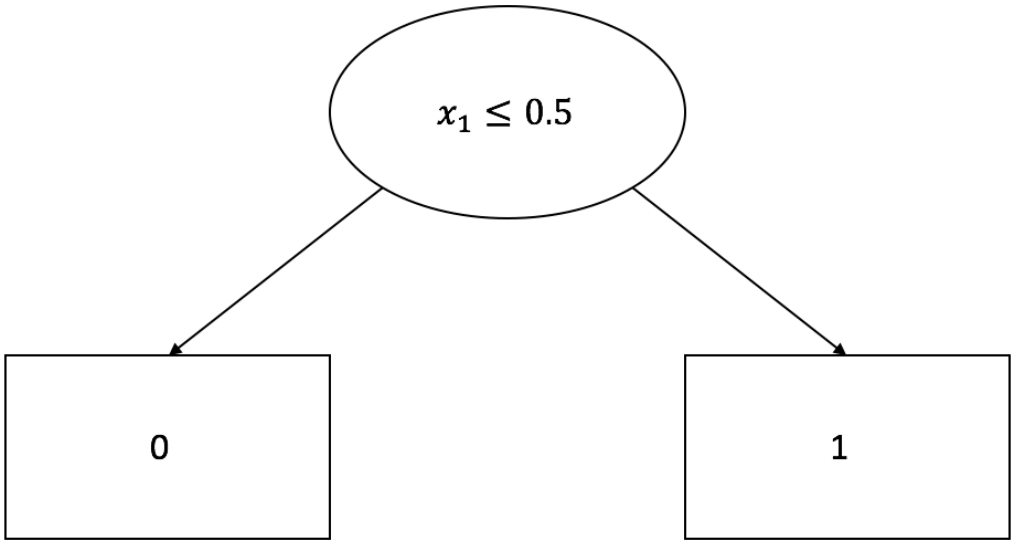
If we split on x2 first, the optimal depth 2 decision tree looks like below:



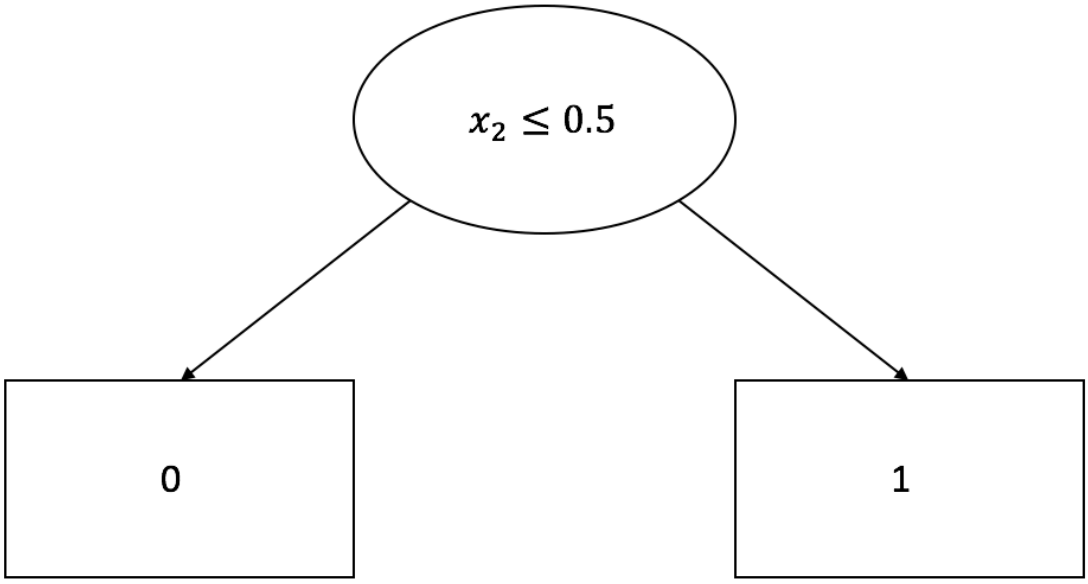
Both of the trees give true risk 0.

2. Variable importance for trees and random forests

(a)/(i) The best split is $x_1 \leq 0.5$ and it gives the reduction in impurity as 0.2703185497750237.



The best surrogate split is $x_2 \leq 0.5$ and $\lambda(1, 0.5, 2, 0.5)$ from Equation(4) equals to 0.6008230452674896.



(a)/(ii)

For equation(2): Imp(x_1) = 0.2703185497750237, Imp(x_2) = NA, Imp(x_3) = NA,

Imp(x_4) = NA, and Imp(x_5) = NA.

For equation(3): Imp(x_1) = 0.2703185497750237, Imp(x_2) = 0.10556222981883365, Imp(x_3) = NA, Imp(x_4) = NA, and Imp(x_5) = NA.

This suggests that x_1 is more important than x_2.

(a)/(iii) The mean least-squares error of predictions on the test dataset of the first decision stump (with the best split $x_1 \leq \mathbf{0.5}$) is 0.1.

The mean least-squares error of predictions on the test dataset of the second decision stump (with the best surrogate split $x_2 \leq \mathbf{0.5}$) is 0.27.

(b)/(i)
k: 1
feature_as_best_split: [202, 204, 179, 218, 197]
feature_as_best_surrogate: [0, 0, 0, 0, 0]

k: 2
feature_as_best_split: [380, 283, 112, 121, 104]
feature_as_best_surrogate: [0, 85, 298, 328, 289]

k: 3
feature_as_best_split: [604, 289, 43, 37, 27]
feature_as_best_surrogate: [0, 313, 229, 0, 458]

k: 4
feature_as_best_split: [820, 180, 0, 0, 0]
feature_as_best_surrogate: [0, 621, 0, 0, 379]

k: 5

feature_as_best_split: [1000, 0, 0, 0, 0]

feature_as_best_surrogate: [0, 1000, 0, 0, 0]

The result suggests x1 and x2 are more important than others. And with the increase of k, the conclusion becomes clearer. When k increases, the best split is more likely to be included in the k randomly selected features to compete for the best split.

(b)/(ii)

k: 1

variable_importance: [0.26889319100980535, 0.10704686421432905, 0.0018217867842835825, 0.0018046335293175762, 0.00147849255977755]

variable_importance_OOB: [0.37173300426068867, 0.22590332168381144, -0.006262930954971552, -0.00637364592402797, -0.011977269611505342]

k: 2

variable_importance: [0.2715258208758446, 0.10658391790302015, 0.002668010445625747, 0.002765739292753901, 0.002066790445424696]

variable_importance_OOB: [0.21674940316140265, 0.11572900087167537, -0.0001902082698179544 4, -0.0057907698812522, -0.0061933228184437306]

k: 3

variable_importance: [0.27261869452382503, 0.10461645674381821, 0.003813689341782229, 0.003804742836831827, 0.0024478748803100758]

variable_importance_OOB: [0.18688053984403366, 0.07425584141045112, 0.003812300076157361, -0.003217113238497782, -0.010937361961698177]

k: 4

variable_importance: [0.27079065933480007, 0.10705449943813226, 'NA', 'NA', 'NA']

variable_importance_OOB: [0.17425703760838665, 0.06838076980308151, 'NA',

'NA', 'NA']

k: 5
variable_importance: [0.27007061090763096, 'NA', 'NA', 'NA', 'NA']
variable_importance_OOB: [0.15695330704588692, 'NA', 'NA', 'NA', 'NA']

The result suggests that variable x1 and x2 are more important than others. When k equals to 5, masking hides the potential variable importance of some variables. When k is less than 5, the impact of masking is lessened.

(b)/(iii)
k: 1
method1: 0.14
method2: 0.37578999999999857

k: 2
method1: 0.14
method2: 0.3584299999999989

k: 3
method1: 0.14
method2: 0.3580499999999991

k: 4
method1: 0.1
method2: 0.3550299999999996

k: 5
method1: 0.1
method2: 0.3316499999999991

Method 1 is correct for computing the prediction error of the random forest. And with the increase of k, the loss becomes lower.

(c)/(i)

q: 0.4

variable_importance: [0.27123453284784244, 0.10762076369087749, 0.002777084730918119, 0.00242164528359711, 0.0019718695404381944]

variable_importance_OOB: [0.23671589200920015, 0.12136986702412476, -0.007080912639175065, -0.003535539997290982, -0.003496071443810864]

q: 0.5

variable_importance: [0.26931120712761447, 0.10422715965660936, 0.003127202584617737, 0.0027933929023339576, 0.0022792461512326804]

variable_importance_OOB: [0.2362724965828761, 0.11809507413807595, -0.004267852903176363, -0.0033951025207189056, -0.008046557123332477]

q: 0.6

variable_importance: [0.2687199407706259, 0.10486185082876386, 0.003598824924252814, 0.0029484052499524605, 0.002330666885330949]

variable_importance_OOB: [0.2198945615933854, 0.10671263977775577, 0.0005008464153941018, -0.007116214292067689, -0.010227554076180331]

q: 0.7

variable_importance: [0.2694841979134777, 0.10531696583241908, 0.0026536851589410585, 0.0027002917668370245, 0.002029181991486937]

variable_importance_OOB: [0.21703663356936842, 0.0958065779616268, -0.0025074445977244964, -0.004325497230782515, -0.007829096629674645]

q: 0.8

variable_importance: [0.2692423017744716, 0.10432061697722969, 0.0029846521495579944, 0.0027382320874146356, 0.0023236349129970566]
variable_importance_OOB: [0.24256205247848286, 0.12207035661873686, -0.009615097394518234, -0.003536605675220361, -0.006230586111931155]

The result suggests that variable x1 and x2 are more important than others. And with the increase of q, the difference of variable importance using equation (6) (OOB) to compute between the best split and the best surrogate split reduces.

(c)/(ii)
q: 0.4
standard deviation of variable_importance: [0.024601876923324287, 0.020152678287313378, 0.002862212154036235, 0.0023014682425255644, 0.0016142712134297505]
standard deviation of variable_importance_OOB[0.22624176784404018, 0.15628171915833503, 0.029298102278023295, 0.02634721493386941, 0.028299227491903545]

q: 0.5
standard deviation of variable_importance: [0.024072083875493765, 0.02086951225716807, 0.003065796704269734, 0.0025846627502915474, 0.002442372883608173]
standard deviation of variable_importance_OOB[0.23714108673150594, 0.16383153074238158, 0.0247803624904617 4, 0.027628703702960177, 0.02492255365352448]

q: 0.6
standard deviation of variable_importance: [0.02588130183302285, 0.018983492768739046, 0.0033776133204824754, 0.0030735034465874194, 0.002338456922416438]

standard deviation of variable_importance_OOB[0.24141675789914452, 0.16252042444694417, 0.03157769799850785, 0.030303832279825603, 0.03260368070520223]

q: 0.7
standard deviation of variable_importance: [0.026145885166509248, 0.020523784766398913, 0.0029037642125973495, 0.002774306000953417, 0.0020124616642561276]
standard deviation of variable_importance_OOB[0.23970784979903836, 0.16608086427392482, 0.027755418424421086, 0.03007334966682433, 0.03387395585585599]

q: 0.8
standard deviation of variable_importance: [0.024234130787151656, 0.020034538109458967, 0.0030975593565063402, 0.0025501574396086944, 0.0019190919550638196]
standard deviation of variable_importance_OOB[0.22202291231619947, 0.1662830068016459, 0.03577605833659686, 0.030234374058633755, 0.025408781284285826]

With the increase of q, the standard deviation of the best split decreases and the standard deviation of other features increases. The reason for this is the more data you put into training, the less likely your classifier will overfit.