# COMS 4771 Machine Learning 2023 Spring
# Problem Set #1

Danrui Wang - dw3031@columbia.edu

2/27/2023

## 1 Designing socially aware classifiers

### Part 0

#### (i)

Because even if we remove the sensitive attribute A, other seems-like nonsensitive attributes $x \in X$ might also potentially correlated with these biased features. For example, when we use ML to predict recidivism, even if we remove race and value attributes like education, education itself might also highly related to race as well because we all know that the fairness of college admission between different races is still an issue today.

### (Part 1)

#### (ii)

$$\begin{aligned} P[\hat{Y}=1] &= P_1[\hat{Y}=1] * P[a=1] + P_0[\hat{Y}=1] * P[a=0] \\ &= P_a[\hat{Y}=1] * (P[a=1] + P[a=0]) \\ &= P_a[\hat{Y}=1] \end{aligned}$$

#### (iii)

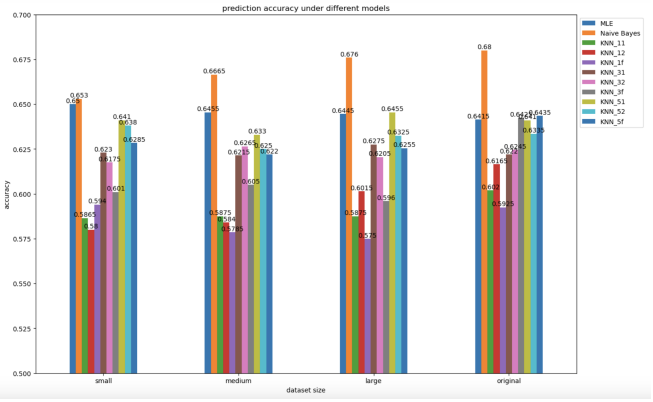
$$\begin{aligned} P[\hat{Y}=\hat{y}] &= P_{a1}[\hat{Y}=\hat{y}] * P[a=a1] + ... + P_{a2}[\hat{Y}=\hat{y}] * P[a=a2] + ... + P_{an}[\hat{Y}=\hat{y}] * P[a=an] \\ &= P_a[\hat{Y}=1] * (P[a=a1] + P[a=a2] + ... + P[a=an]) \\ &= P_a[\hat{Y}=1] \end{aligned}$$

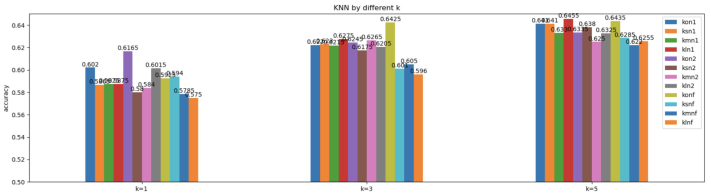
The generalized form is:

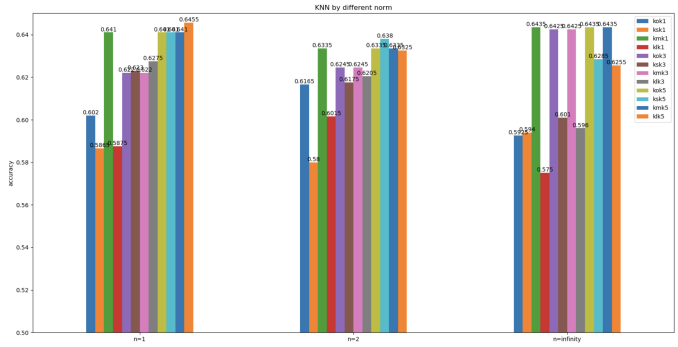
$P_i(\hat{Y}=\hat{y}) = P_j(\hat{Y}=\hat{y}) \iff P[\hat{Y}=\hat{y}] = P_a[\hat{Y}=\hat{y}]$ for i, j, a $\in A, A \in \mathbb{N}, \hat{y} \in Y, Y \in \mathbb{R}$

# Part 2

## (v)




I randomly sampled the training dataset into different sizes, with small = 1000, medium=2000, large=3000m and the original dataset = 4167. For knn, I combined k = 1,3,5 with norm = 1,2 and infinity. In the graph representation knn_3f for example, the first digit 3 represents the k=3 and the second digit f indicates the number of norm = infinity.

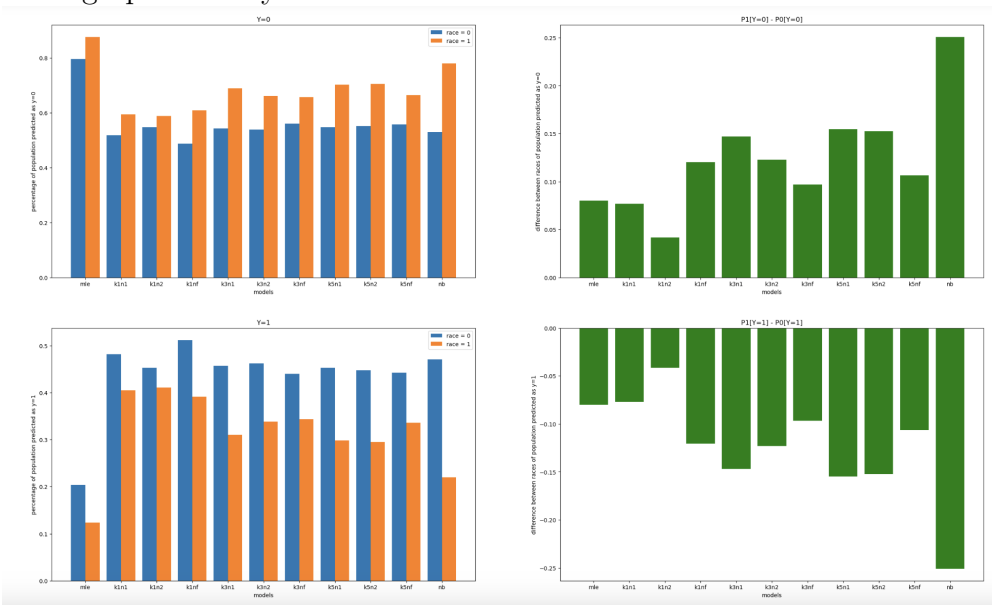
According to the graph, naive bayes model performs the best across all size of data, and the larger the dataset, the higher the accuracy achieved; the highest accuracy among all is 0.68. As for MLE, the performance decreases as the dataset size increases (but if I resample the small, medium, and large dataset, this relationship might be gone). The highest accuracy achieved by MLE with the small dataset is 0.65. As for KNN, the accuracy of knn with all k and norm does not show specific relationship between different dataset sizes. However, the prediction is generally more accurate as k increases. Some combination of k and data size works better with norm = 1 and infinity, some works best with norm = 2, and also some works the worse with norm = infinity. The highest accuracy achieved in KNN model is 0.6455, with k=5, norm = 1, and under large dataset.

(* Because I randomly sample the small, medium, and large dataset, outputs for each new run might be different. I also scale the factors, but the accuracy is lower than non scaling, so I choose the output without scaling.)

**(vi)**

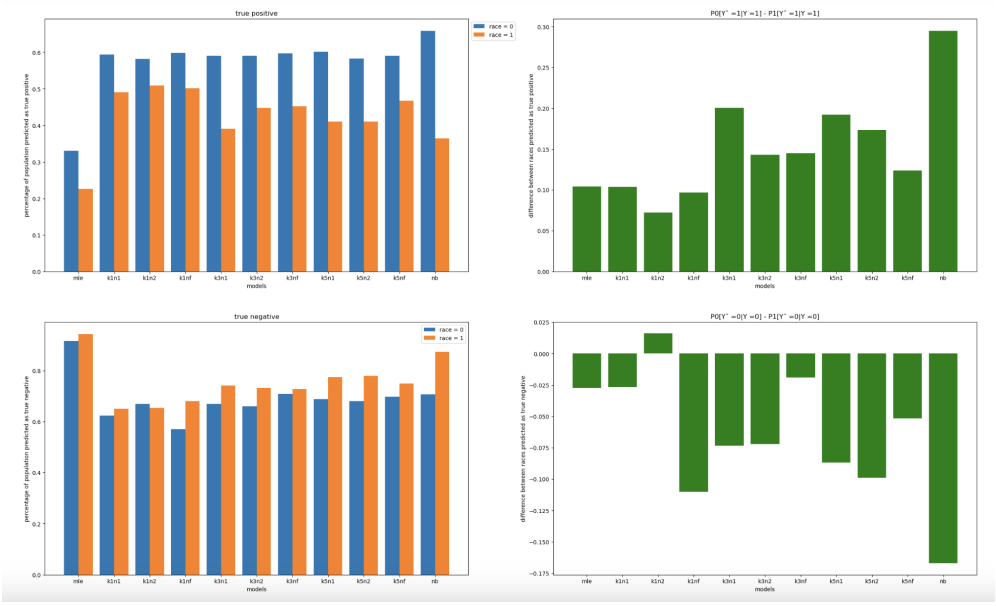
Because we didn't see direct or obvious relationship between dataset size and accuracy and I've tested in fairness for mle which also didn't show obvious impact of dataset size on fairness, for this question, I would test all models with the original dataset.

Demographic Parity:



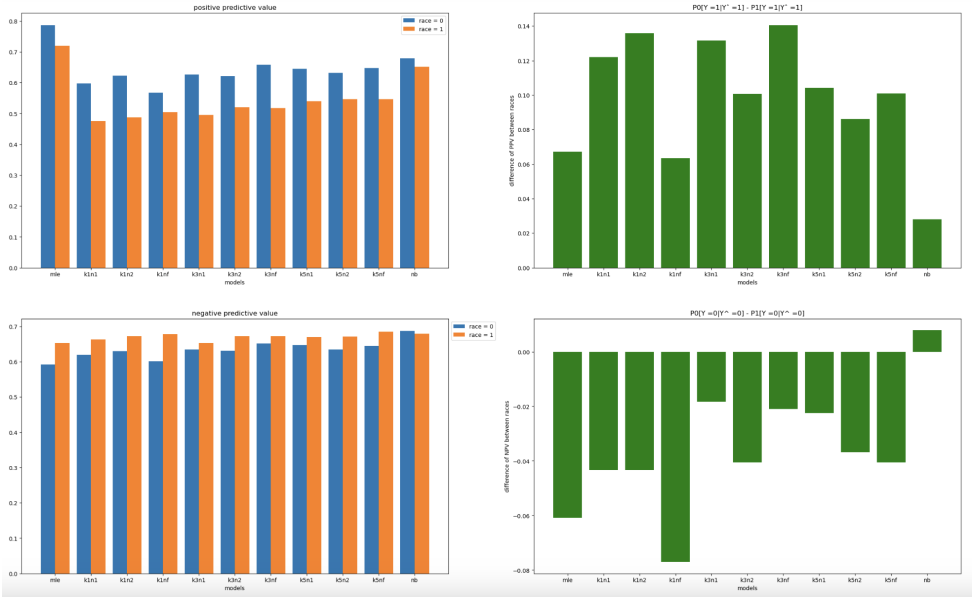

For all models, race=1 are more likely to predict two_year_ recid = 0, and race=0 predicts more to have two_year_ recid = 1. Knn with k=1 and n=2 is the fairest model, and naive bayes achieves the worst.

Equalized Odds:




For all models, race=0 gets much more true positive than race=1. True negative by comparison is more fair between races than true positive. For both true positive and true negative, knn with k=1 and norm=2 is the fairest, and naive bayes is also the worst.

Predictive Parity:




Similar to equalized odds, for all models, race=0 gets more positive predictive values than race 1. Even though race=1 gets more negative predictive values, the difference shrinks in negaitive predicitive values between races. For both models, naive bayes is the fairest. KNN with k=3 and norm = infinity is the worst for PPV, and KNN with k=1 and norm = infinity is the worst for NPV.

**(vii)**

When a company hires people, if we consider about gender, demographic parity ensure that the percentage of each gender being hired are equal. However, it does not ensure the standard that those people being hired are equal, which means it does not make sure $P_0[\hat{Y} = \hat{y}|Y = y] = P_1[\hat{Y} = \hat{y}|Y = y]$. For example, the company can hire 50 percent of male workers who do not have the ability to gain the job, but hire 50 percent of female workers who meets the standard.

# 2 Data dependent perceptron mistake bound

## (i)

Since $T_\gamma \leq w^T w* \leq ||w^T||||E*|| \leq R\sqrt{T}$, if we want to proof the mistake bound is tight which makes exactly $((\frac{R}{\gamma})^2$, we need to show $T_\gamma = w^T w^* = ||w^T||||w^*|| = R\sqrt{T}$. We can show this using the dataset containing (0,1) and (0,-1), around the sphere which has R=1. Two lines are at the tangent of the sphere and parallel to each other. Then for iterations t we have,

$$\begin{aligned} \gamma &= yxw^* \\ &= ycos\theta||x||||w^*|| \\ \frac{\gamma}{||w^*||} &= ycos\theta||x|| = \gamma \\ w^t w^* &= (w^{t-1} + yx)w^* \\ &= w^{t-1}w^* + \gamma \end{aligned}$$

Since in other dataset, all projection have the same length,

$$||w^t||^2 = ||w^(t-1)||^2 + 2y(w^(t-1)x) + ||yx||^2$$

Where $2y(w^(t-1)x) \leq 0$ Thus, when $2y(w^(t-1)x) = 0$, $R = ||x||$, meaning $max_{x_i \in S}||x_i|| = ||x||$

## (ii)

$$\begin{aligned} ||w_T||^2 &\leq ||w_{T-1}||^2 + ||x_{i_T}||^2 \leq ||w_{T-1}||^2 + ||(I-P)x_{i_T} + Px_{i_T}||^2 \\ &= ||w_{T-1}||^2 + ||(I-P)x_{i_T}||^2 + ||2(I-P)x_{i_T}Px_{i_T}|| + ||Px_{i_T}||^2 \end{aligned}$$

Since (I-P) is the projector onto the orthogonal complement space of w*, P is orthogonal to (I-P)

$$\begin{aligned} ||w_T||^2 &\leq ||w_{T-1}||^2 + ||(I-P)x_{i_T}||^2 + ||Px_{i_T}||^2 \\ &\leq \epsilon^2 + ||w_{T-1}||^2 + ||Px_{i_T}||^2 \end{aligned}$$

Thus after T iterations,

$$||w_T||^2 \leq \epsilon^2 T + \sum_{t=1}^{T} ||Px_{i_T}||^2$$

## (iii)

$$w_T = w_{T-1} + y_{i_T} x_{i_T} = \sum_{t=1}^{T} y_{i_t} x_{i_t}$$

$$\begin{aligned}(w_T \cdot w^*)^2 &= (\sum_{t=1}^{T} y_{i_t} x_{i_t} \cdot w^*)^2 \\ &= \sum_{t=1}^{T} (y_{i_t} x_{i_t} \cdot w^*)^2 + 2 \sum_{j=1}^{T-1} \sum_{k=j+1}^{T} (y_i j) xij \cdot w^*)(y_{ik} x_{ik} \cdot w^*) \\ &= \sum_{t=1}^{T} (x_{i_t} \cdot w^*)^2 + 2 \sum_{j=1}^{T-1} \sum_{k=j+1}^{T} (y_i j) xij \cdot w^*)(y_{ik} x_{ik} \cdot w^*)\end{aligned}$$

Since $\forall i$, $y_{i_t} x_{i_t} \cdot w^* \geq \gamma$, and since $Px_{i_t}$ is the projection of $x_{i_t}$ on w*, meaning $||Px_{i_t}|| = x_{i_t} \cdot w^*$,

$$(w_T \cdot w^*)^2 \geq T(T-1)\gamma^2 + \sum_{t=1}^{T} ||Px_{i_t}||^2$$

## (iv)

Because w* is a unit vector with length 1,

$$\begin{aligned}T(T-1)\gamma^2 + \sum_{t=1}^{T} ||Px_{i_t}||^2 &\leq \epsilon^2 T + \sum_{t=1}^{T} ||Px_{i_T}||^2 \\ T(T-1)\gamma^2 &\leq \epsilon^2 T \\ T &\leq (\frac{\epsilon}{\gamma})^2 + 1\end{aligned}$$

# 3 Constrained Optimization

minimize: $||x - x_a||^2$
constraint: $g(x) = wx + w_0 = 0$

Since $||x - x_a||^2$ and g(x) are both convex, this is a convex optimization problem.

$$L = ||\overrightarrow{x} - x_a||^2 - \lambda(w\overrightarrow{x} + w_0)$$

$$\frac{\partial L}{\partial \overrightarrow{x}} = 2(\overrightarrow{x} - x_a)^T - \lambda w = 0$$

$$\overrightarrow{x} = \frac{2x_a + \lambda w^T}{2} = x_a + \frac{\lambda}{2}w^T$$

$$\frac{\partial L}{\partial \lambda} = -w\overrightarrow{x} - w_0 = 0$$

$$-w(x_a + \frac{\lambda}{2}w) - w_0 = 0$$

$$-wx_a - \frac{\lambda}{2}ww^T - w_0 = 0$$

$$-\frac{\lambda}{2}ww^T = w_0 + wx_a$$

$$\lambda = -\frac{2(w_0 + wx_a)}{ww^T}$$

if $||w|| \neq 0$

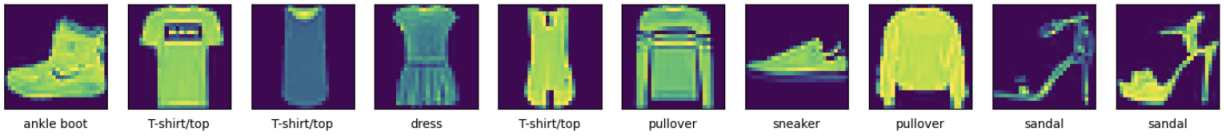
Hence,

$$||\overrightarrow{x} - x_a||^2 = ||x_a - \frac{w_0 + wx_a}{ww^T}w^T - x_a||^2$$
$$= \frac{(w_0 + wx_a)^2}{||w||^2}$$

Since if $||w||^2 = 0$, in order to satisfy the constraint, $w_0 = 0$. In this case, g(x) is no longer a hyperplane, so we do not consider this case.

# 4 Decision Trees, Ensembling and Double Descent

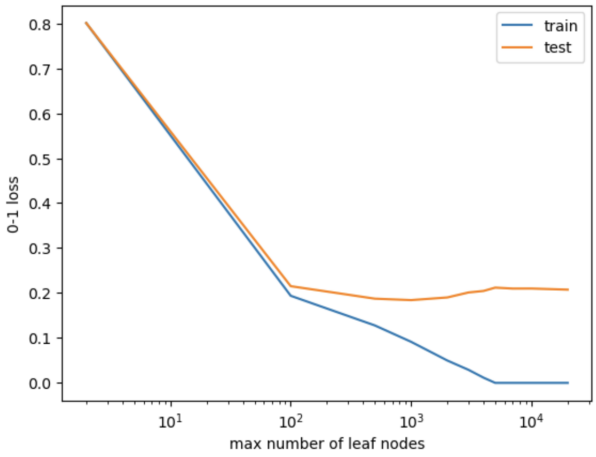
## (i)




Fashionmnist is harder than classical MNIST, because classical MNIST classifies hand written number. The MNIST images are simpler which some model can even predict by one pixel (Franceschini). However, images in FashionMNIST is denser, which makes them harder to be classified. As for KNN, KNN does not learn anything and it doesn't train on the training dataset, KNN only outputs the labels belong to most of its neighbors. Also, since each clothes looks different, looking for the distance between each pixel might not be a good choice.

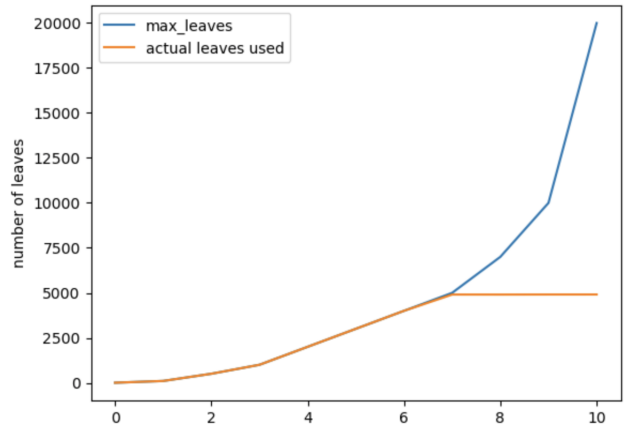
Reference: Franceschini, Luca.Distinguishing pairs of classes on MNIST and Fashion-MNIST with just one pixel. April, 2021. https://lucaf.eu/2021/04/05/mnist-pairwise-one-pixel.html

## (ii)




The minimum loss is 0.1846. The training and test 0-1 loss were almost the same until log max number of leaf nodes = $10^2$. From then, the training loss keeps decreasing, while testing loss deacrease a little bit and start to rebound. This range might induce overfitting problem. When getting closer to $10^4$ , both training and test 0-1 loss stop changing and became parallel. At this range, the training loss is 0. If I don't set the $max_numleaf$ limit, the model uses 4906 leaves, which looks like the turning point of both training and loss before stop decreasing.

## (iii)



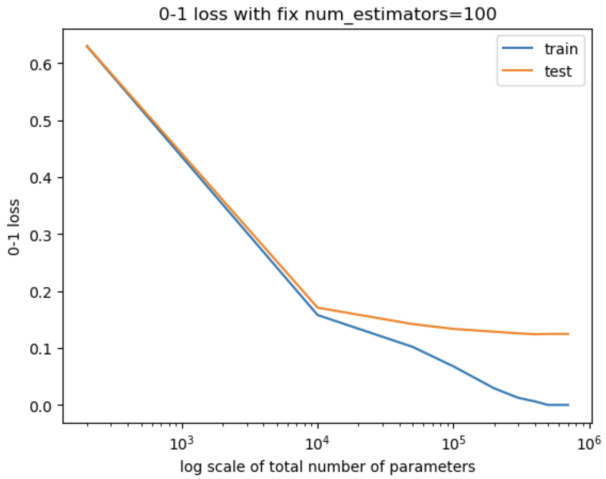

No. The maximum number of leaves used is 4908. After that, whatever the maximum limit I set, the model would not use more leaves than 4908.

## (iv)

Because in random forest, we train a series of decision trees on random subsets of the original, this means that the size of each subset is much smaller. If the original training dataset has a lot of features and if we include all these features, the smaller subset might be overfitted, which leads to high variance. Thus, in order to reduce the variance, we can only include a subset of all features when training each ransom subsets.

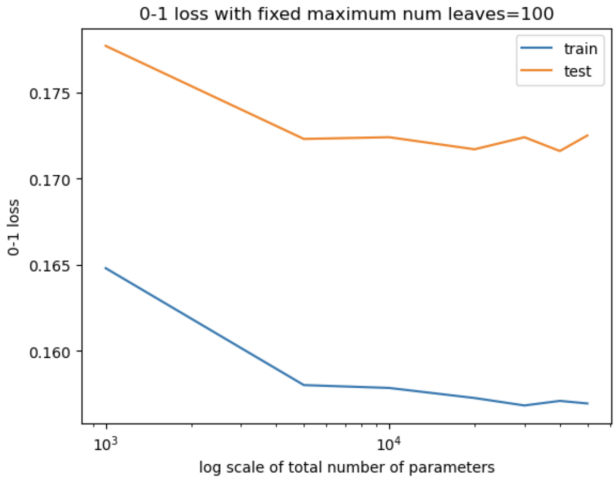
## (v)

### (a)



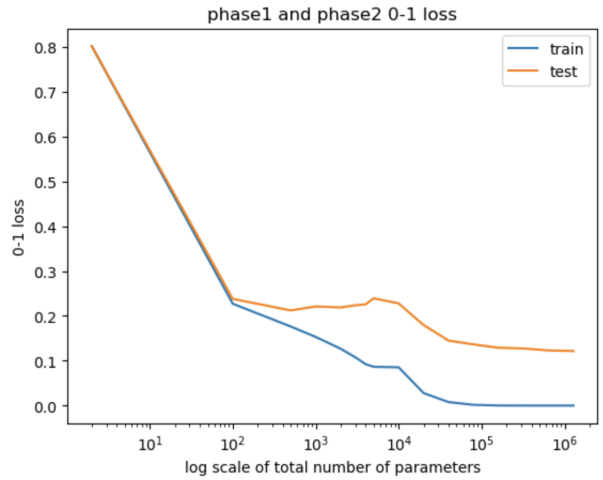

I set the $num_estimators$ as default = 100. As the log scale of total number of parameters approaches $10^6$, the 0-1 loss of training set approaches 0, and the 0-1 loss of test dataset stopped decreasing. This might lead to overfit problem.

## (b)




I set the maximum number of leaves to be 100, which is much smaller compared to (iii). The best loss approaches 0 but not equal to 0 in this case. This means that we are less probable to have overfitting problem compared to decision tree. Compared to graph in (a), with similar number of parameters, the loss is generally much smaller for fix max number of leaves=100 and increase the number of estimators. However, unlike (a) where there is a range train and test have the same 0-1loss, in this case, train and loss are almost always parallel, where test loss is always larger than train loss.

## (vi)




After phase1, both training and testing 0-1 loss re-bounce and drop further during phase 2. As before, the surprising point is ones again around 4900 which is the maximum number of leaves. Also, generally, before the surprising point, the loss decreases very fast before the surprising point if we only increase the maximum leaves permitted. If we only increase the number of estimators after this point and increase the overall number of parameters, the drop would be slower. Also, after phase 1, if we divide the original dataset into subgroups, we see furthure loss drop.