THEORETICAL ASSIGNMENT

True or False(Mention reason)

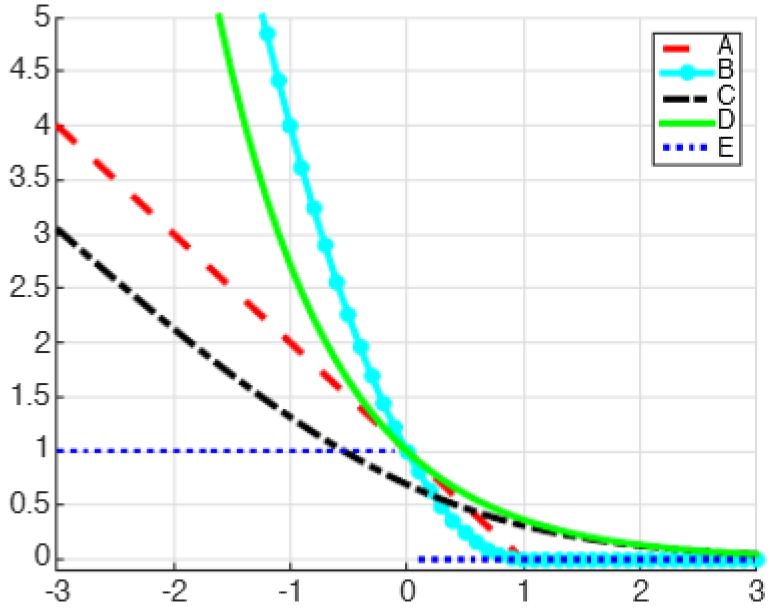
1. Decreasing the depth of your decision tree (through pruning) will reduce test error.
2. In k-fold cross validation you leave kinputs out, train your classifier on the remaining n−k inputs and evaluate it on the leave-out inputs. You do this repeatedly and average to obtain a good estimate of your classifier's performance.
3. Gradient Boosting is performing (stage-wise) gradient descent in function space
4. When you split your data into train and test you have to make sure you always do the splitting uniformly at random.
5. If a classifier obtains 0% training error it cannot have 100% testing error.
6. Increasing regularization tends to reduce the bias of your classifier.
7. If run without a depth limit, the ID3 algorithm returns the maximally compact decision tree that is consistent with a data set (if it exists).
8. The best classifiers make no assumptions about your data at all.
9. Random Forests learn many high variance CART trees and reduce this variance by averaging the results. That's basically Bagging applied to (slightly modified) CART trees.
10. As your training data set size, n, approaches infinity, the k−nearest neighbor classifier is guaranteed to have an error no worse than twice the Bayes optimal error.
11. Squared loss regression trees require a time complexity O(n2) per split.
12. The Bayes optimal error is the best classification error you could get if ther was noise.

Answers:

1) **True**, it would reduce the test error as the whole purpose of pruning is to remove the branches with significantly less influence, hence prevent overfitting and increasing the accuracy of the model
2) **True,** we make n/k subsets of the data and then repeatedly train the model on (n/k) - 1 subsets and test on the remaining subsets while varying the subset we have left out.
3) **True.** it combines multiple weak learners to create one strong learner and then builds a new weak learner to minimize the gradient
4) **True,** in order to maintain the randomness of the datasets and avoid any patterns that the model can learn apart from those that it gains from the data features
5) **False,** there can be an extreme case scenario where the model is extremely overfitted on the training dataset and has memorized all the data points and hence can produce a very high amount of error on the test set
6) **False,** Bias is the error due to very simple assumption while variance error due to the model learning noise, regularization is meant to reduce overfitting and by extension variance not bias.
7) **False,** When an ID3 algorithm runs without a depth limit, it will continue to split the data until each leaf has examples of only a class which leads to very deep tree that won't be maximally compact
8) **True**
9) **True**
10) **True,** Thomas Cover and Peter Hart in their 1967 paper "Nearest Neighbor Pattern Classification" that for a large dataset, i.e n approaches infinity, the error is at most twice the Bayes optimal error
11) **False,** they require a time complexity of O(n*log(n))

12) **True,** Bayes optimal error is defined as the lowest possible error a model could have, i.e the error of a near perfect model

ERM and SVM



1.(3) Write down the loss and regularizers of SVM,LASSO and Ridge Regression.
Provide names for all loss functions and regularizers.
2.(2.5) Match the loss to the figure (just write A, ..., E next to each loss function's name.)
3.(3.5) Which loss functions in Q2 could you minimize with Gradient Descent without modifications? Which could you minimize with Newton's Method? Justify your claims if you cannot use an approach for a given loss functions (no justification needed if a method does apply)

Answers:

1) SVM: Loss function: Hinge Loss, Regularization: L2 Regularization
LASSO: Loss function: MSE, Regularization: L1 Regularization
Ridge regression: Loss function: MSE, Regularization: L2 Regularization
2) A) Hinge loss
B) MSE
C) Cross Entropy
D) Exponential loss
E) 0-1 Loss
3) Gradient descent is a first order optimization algorithm, and hence can be applied to hinge loss (using subgradients), MSE, Exponential loss and cross entropy, it cannot be applied to 0-1 loss as the function is not differentiable throughout.

Newtons's method can be applied to cross entropy, MSE, and exponential loss. It cannot be applied to hinge loss or 0-1 loss as it isn't second order differentiable,

Bias and Variance

1.(3) Scrooge McDuck trains a classifier, trying to predict stock market prices. He trains decision trees of limited depth d (as he wants to reduce CPU time and electricity cost). However, soon he realizes that his training and his testing errors are both much too high for his system to be useful.Name two possible explanations for what could be the root of the problem.
2. (3) Scrooge now decides to no longer limit the tree depth. Instead he trains trees with unlimited depth. To his big disappointment he observes very similar behavior (train and test error are too high). What explanation can you give him?
3. (6) What does Bagging simulate and why would it reduce variance?
4. (3) You boost decision trees with very limited depth (depth = 2). How are bias and variance affected as the boosting iterations increase.

Answers:

1. The error in his model could be extremely high due to underfitting as the model is somewhat shallow due to the limited depth. Furthermore it could be due to the bad quality of data, too much noise, and irrelevant features. Lastly, decision trees wouldn't be the best model to predict stock prices.
2. With an unlimited depth tree, the model is bound to be overfitted to the training data memorizing all the noise as well and hence not performing as well on the testing dataset. But since the model has high training errors still, it is more probable that the type of model chosen to predict stock prices is incorrect and that the data is not as clean, with a lot of noise and various irrelevant features.
3. Bagging stimulates creating multiple datasets from the original dataset and training a model on each of these different samples. To get the final prediction, we aggregate the results of the various models. For example, for a regression model, we average the outputs while a classification system uses voting.
   It reduces variance as by forcing the model to train on different samples makes the model focus on different aspects of the data. Furthermore, by averaging out the predictions of the model we reduce the variance of the model.

kNN / Curse of Dimensionality

1. (3) Kim K. uses kNN classification with the Euclidean distance.
She has n= 100000 data points in her training set, each with
d= 50 dimensions. She is frustrated, because during test-time her classifier is
too slow. What would you recommend her to do in order to speed up her
classifier during test time?

2. She completely ignores your advice and instead analyzes the code herself. She
realizes that most of her computation time is spent computing the √operator.
Annoyed by this wasted CPU power, she decides to use the squared distance
instead, [dist(~x, ~z)]2(and drops all square-root computations). Although faster
to compute, this squared Euclidean distance is no longer a metric, as it does not
satisfy the triangular inequality.
a) (3) Will this missing property affect her kNN accuracy? Explain why/ why
not.
b) (3) If she is determined to use the squared distance, is your answer to question 1 still valid?
Explain why/why not.

3. (3) Kanye W. wants to use the kNN classifier on images of dresses to classify
them as either white with gold stripes (+1) or blue with black stripes (-1). The
data set is rather high dimensional d= 1000000 (each feature is a pixel) and
he only has about n= 5000 images. Should he be worried about the curse of
dimensionality? Explain why/why not.

4. (2) Explain how the neighborhood size k affects the classifier's bias and variance.

5. (3) Provide one scenario in which you would want to use a kNN classifier instead
of a linear SVM classifier and vice versa.

Answers:

1) The methods to speed her classifier could be:
    a) Data Preprocessing: Normalizing all the features to be between 0 or 1 which would save computational resources and we could select the most relevant features to save time.
    b) Reduce the value of k which would reduce time by having to calculate lesser number of euclidean distances. But this might affect the accuracy of the classification.
2) It will be affected as follows:
    a) This missing property will not affect kNN accuracy, as it depends only relative comparison of distances, so if d1 > d2 then (d1)^2 > (d2)^2 will also hold true.
    b) Yes the answer to question 1 is still valid as using the squared distance with those modifications would still make the same difference
3) The problems faced would be:
    a) As the dimensionality increases, the distances become very similar making it very hard for the model to separate between the data points.
    b) Due to the low number of datapoints, and such high dimensionality, it will become even harder for the model to form meaningful neighborhoods and it would become easy for the model to become overfitted on the training data.

        c) The computational cost would increase by a very large amount due to the very high number of dimensions.
4) Ans
    a) Small K: This would lead to a low bias and high variance since the model would be trained to be fitted to various data. This would lead to low bias but at the same time it would lead to high variance as it would fit a lot of noise as well. It would be very sensitive to a few outliers as well.
    b) High K: This would lead to low variance but high bias. With a large K, it would make a larger set of neighbors which would lead to it averaging out more number of points and hence having a higher bias. This could lead to the model missing some of the finer relations and details. But at the same time, due to averaging a larger number of data points which makes it less sensitive to outliers.
5) An example where I would use SVM would be in the classification of emails as spam or not spam as the data points are linearly separable. A situation where I would prefer kNN would be in a situation where we are classifying faces based on a variety of features which would lead to a complex, non linear type of classification.

Decision Trees

1. (3) What is the prediction value at a leaf of a regression tree (with squared-loss impurity)? Prove that this is optimal.
(2) Given the definition of the Gini Index of a set:

$$G(S) = \sum_{k=1}^{c} p_k(1 - p_k)$$

where cis the total number of classes, and pk is the probability that an item of set Sis of class k. In the situation where there are 3 classes, when is G(S) maximized, when minimized? (no derivation necessary)?
3. (2) Explain in what sense decision trees are “myopic”.
4. (2) What are two methods of preventing overfitting in decision trees?

Boosting and Bagging

1. (4) Ludwig van Beethoven claims that when he trains a Random Forests classifier he does not need any validation or test data and can train the classifier on the entire data set, even to select model parameters? Is this true/false? Justify your answer.
2. (5) What is the loss function that Adaboost minimizes? Prove that it is an upper bound on the training error.

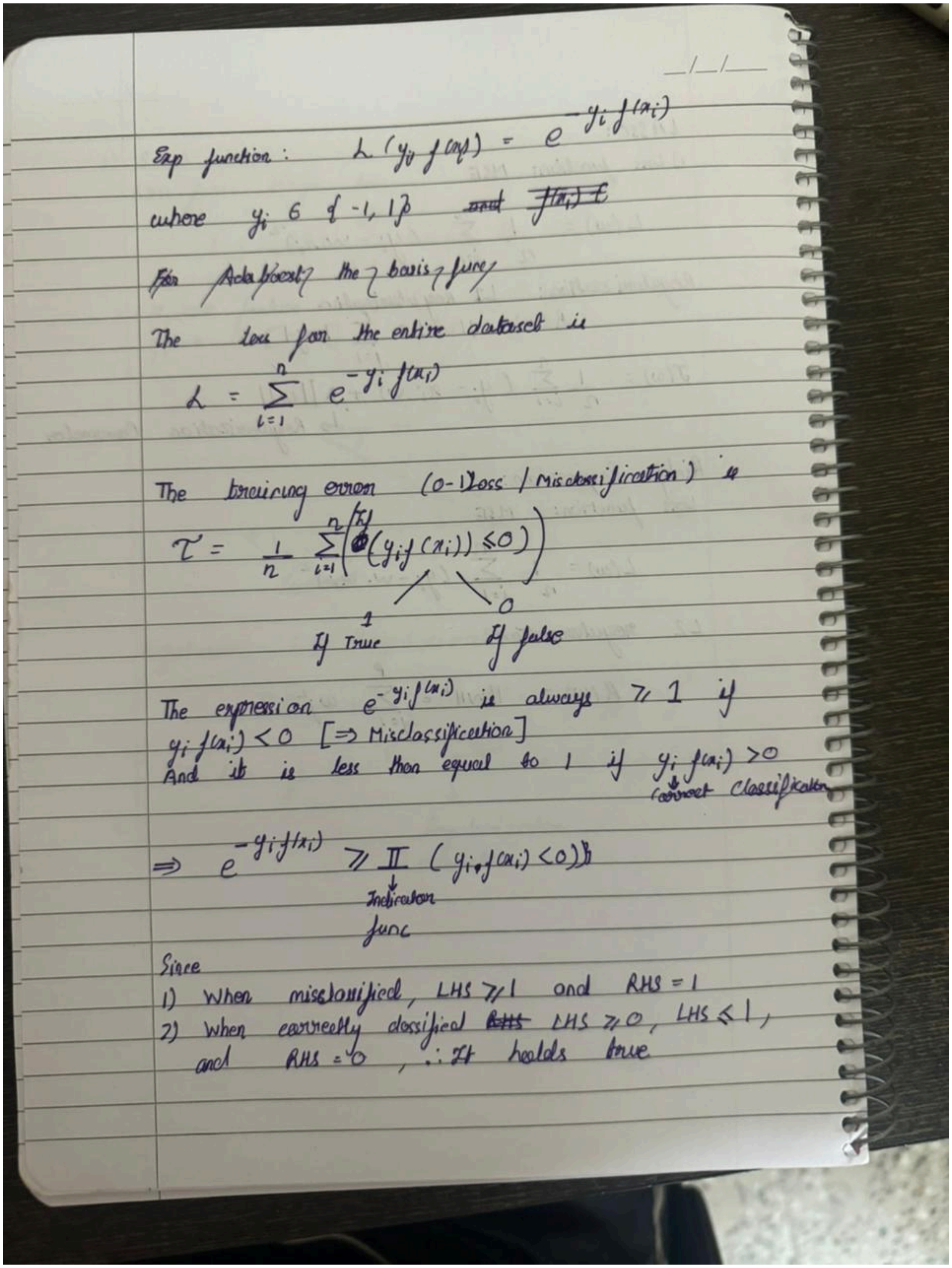
Answers:
1) This claim is false as even with random forests we need validation and test data to ensure:
    a) That the model isn’t overfitted to the training data

b) Correctly estimate the performance of the model
c) Select the correct model parameters using hyperparameter tuning

If we continue to use only the training data even for the hyperparameter tuning, it is more likely that we select parameters specific only to the training data leading to an overfitting to the training data. Furthermore, with no robust method of verifying the models performance, we cannot gauge how it will perform in a random dataset

2) The loss function that Adaboost minimizes is the exponential loss function the proof is as follows: (Photos Below)

3)



Exp function: $L(y_i, f(x_i)) = e^{-y_i f(x_i)}$

where $y_i \in \{-1, 1\}$ ~~and~~ ~~$f(x_i) \in$~~

~~For AdaBoost the basis func~~

The loss for the entire dataset is

$$L = \sum_{i=1}^{n} e^{-y_i f(x_i)}$$

The training error (0-1 Loss / Misclassification) is

$$\tau = \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{I}(y_i f(x_i) \leq 0) \right)$$

1 If True, 0 If false

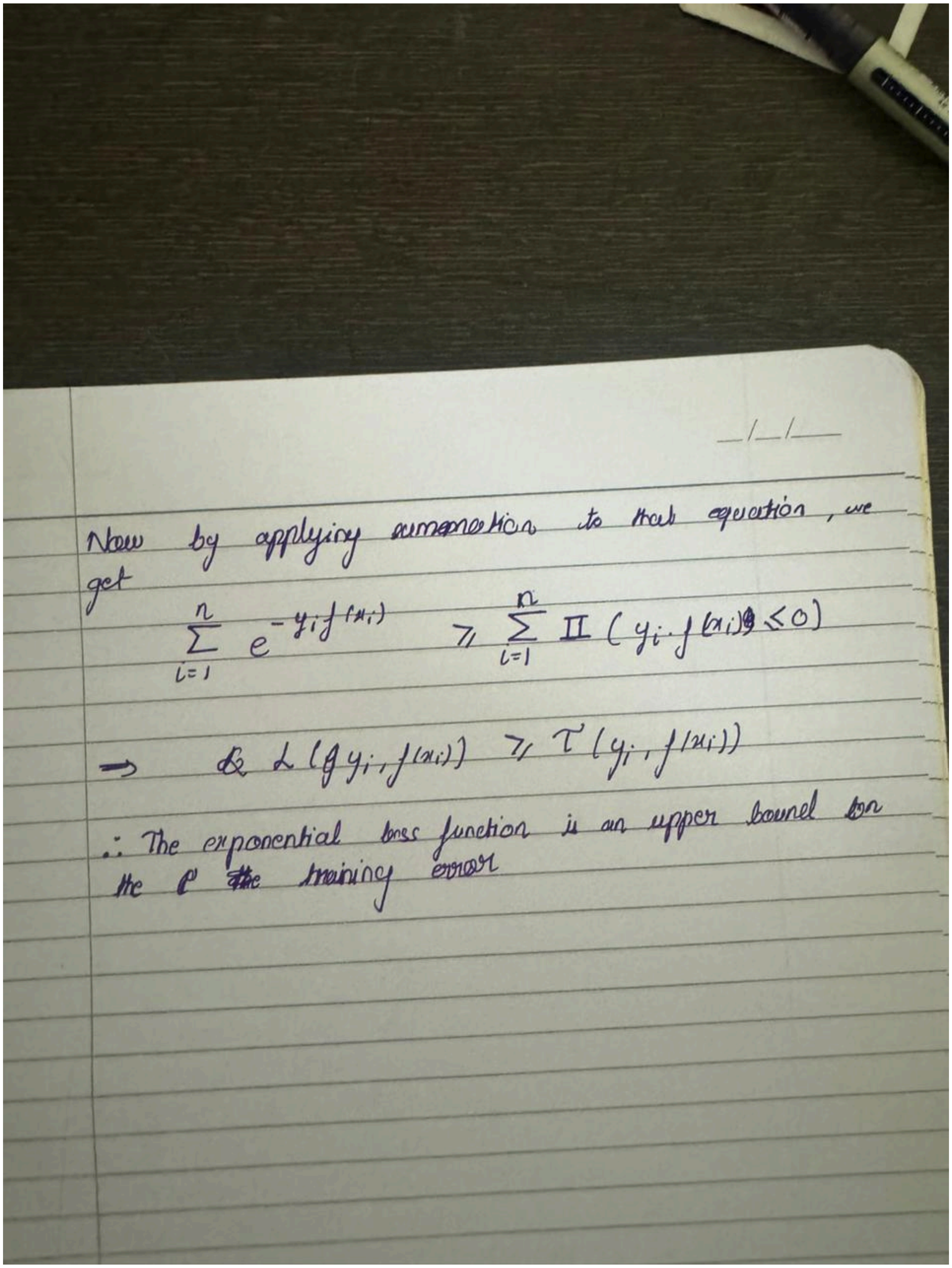
The expression $e^{-y_i f(x_i)}$ is always ≥ 1 if $y_i f(x_i) < 0$ [$\Rightarrow$ Misclassification]

And it is less than equal to 1 if $\underset{\substack{\downarrow \\ \text{correct classification}}}{y_i f(x_i) > 0}$

$$\Rightarrow e^{-y_i f(x_i)} \geq \underset{\substack{\downarrow \\ \text{Indicator} \\ \text{func}}}{\mathbb{I}} (y_i f(x_i) < 0)$$

Since

1) When misclassified, LHS ≥ 1 and RHS $= 1$
2) When correctly classified ~~RHS~~ LHS ≥ 0, LHS ≤ 1, and RHS $= 0$, $\therefore$ It holds true



__/__/____

Now by applying summation to that equation, we get

$$\sum_{i=1}^{n} e^{-y_i f(x_i)} \geq \sum_{i=1}^{n} \mathbb{I}(y_i \cdot f(x_i) \leq 0)$$

$$\Rightarrow \quad L(y_i, f(x_i)) \geq \tau(y_i, f(x_i))$$

$\therefore$ The exponential loss function is an upper bound for the ~~e~~ ~~the~~ training error