Warm-up on *k*-NN, Decision Trees, Random Forests

1. Why is the *k*-NN algorithm called a 'lazy-learner'?

   Solution: Because no computations are performed during the "training step" of the algorithm (there is no training phase). The training classes are stored in memory and final computations/predictions are performed only when an unseen class is introduced.

2. Which of the following distance metrics can not be used for *k*-NN?
   A) Manhattan
   B) Minkowski
   C) Tanimoto
   D) Jaccard
   E) Mahalanobis
   F) All of the above

   Solution: All of the above

3. What are the assumptions/prerequisites of the *k*-NN algorithm?

   Solution: *k*-NN is non-parametric; it does not make any assumptions about the underlying distribution of the data. As such, there are no prerequisites for this algorithm.

4. Which of the following distance measure do we use in case of categorical variables in *k*-NN?

   (i) Hamming Distance
   (ii) Euclidean Distance
   (iii) Manhattan Distance

   A) (i)
   B) (ii)
   C) (iii)
   D) (i) and (ii)
   E) (ii) and (iii)
   F) (i), (ii), and (iii)

   Solution: Hamming Distance

5. What would be the relation between the time taken by 1-NN,2-NN,3-NN:

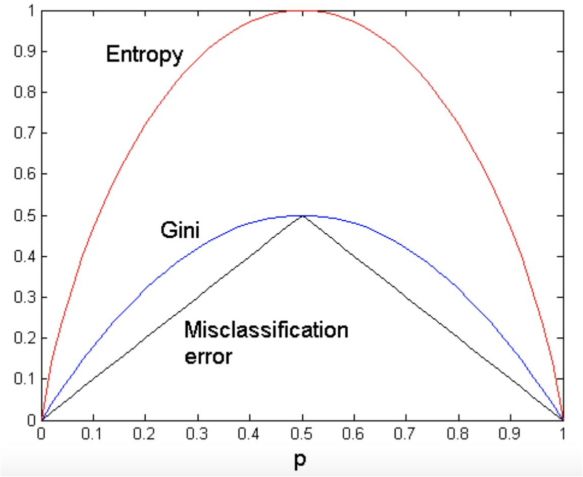
   A) 1-NN >2-NN >3-NN
   B) 1-NN < 2-NN < 3-NN
   C) 1-NN = 2-NN = 3-NN (approx.)
   D) None of these

   Solution: 1-NN = 2-NN = 3-NN (approx.)

6. Discuss the differences and similarities between the following in groups:

   (i) GINI Impurity
   (ii) Entropy
   (iii) Misclassification Error

   Solution:



   From the above image, we see that all three measures perform more or less the same, except at the extreme values of *p*; entropy outperforms Gini and misclassification error at these points.

7. What are the limitations of information gain?

   Solution: Information gain tends to prefer splits that result in large number of partitions, each being small but pure.

8. How do you fit decision trees in the presence of missing values?

   Solution: We re-weight the classes based on the probability of the non-missing

values and compute our decision trees as usual.

9. What is the difference between re-substitution error and generalization error? Discuss in groups.

   Solution: Re-substitution error = Training Error
   Generalization error = Testing Error

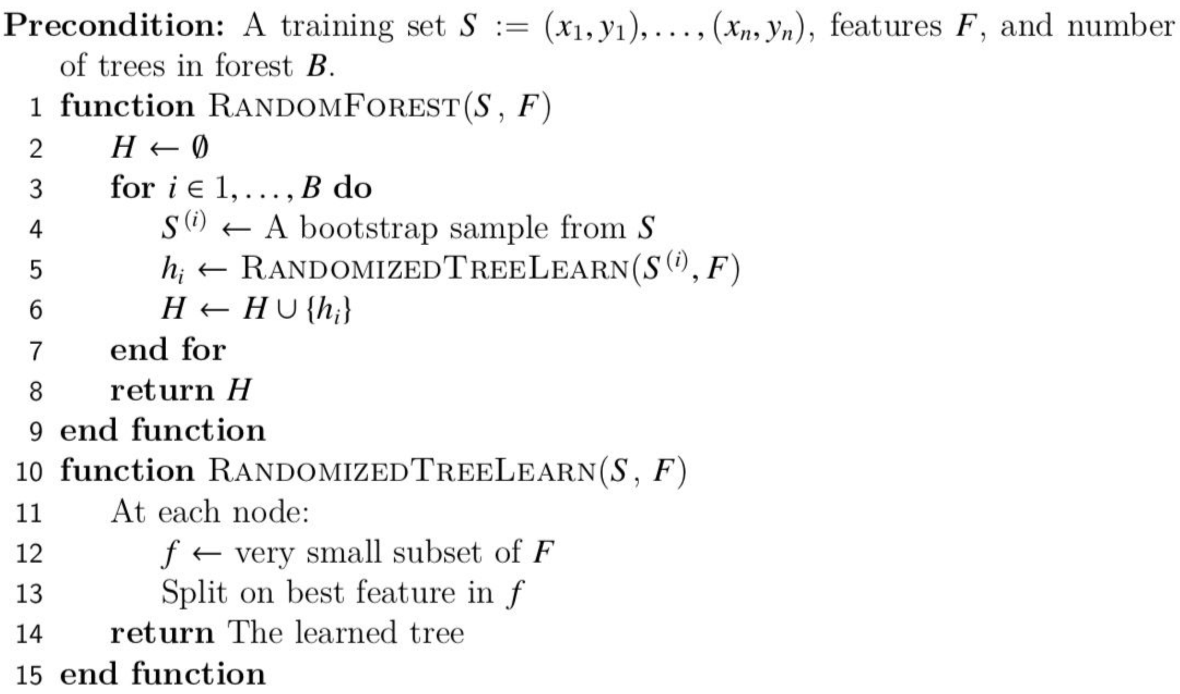
10. What is the difference between rule-based classifiers and instance-based classifiers?

    Solution: Rule-based classifiers identify/learn from if-then conditions and store these results to predict classes. For e.g., association rules, decision trees

    Instance-based classifiers, on the other hand, do not learn underlying associations between data and provide generalizations, but instead, store the training records in memory and use the training records to predict the class label of the unseen class. For e.g., *k*-NN

11. Write a pseudo-code for the random forest algorithm.

    Solution:
    For each tree in the forest, we select a bootstrap sample from *S* where *S* (*i*) denotes the *i*-th bootstrap.
    We then learn a decision tree using a modified decision-tree learning algorithm.
    The algorithm is modified as follows: at each node of the tree, instead of examining all possible feature splits, we randomly select some subset of the features *f* ⊆ *F*.
    where, *F* is the set of features.
    The node then splits on the best feature in *f* rather than *F*.



Precondition: A training set $S := (x_1, y_1), \ldots, (x_n, y_n)$, features F, and number of trees in forest B.

```
function RandomForest($S$, $F$)
    $H \leftarrow \emptyset$
    for $i \in 1, \ldots, B$ do
        $S^{(i)} \leftarrow$ A bootstrap sample from $S$
        $h_i \leftarrow$ RandomizedTreeLearn($S^{(i)}$, $F$)
        $H \leftarrow H \cup \{h_i\}$
    end for
    return $H$
end function
function RandomizedTreeLearn($S$, $F$)
    At each node:
        $f \leftarrow$ very small subset of $F$
        Split on best feature in $f$
    return The learned tree
end function
```

http://pages.cs.wisc.edu/~matthewb/pages/notes/pdf/ensembles/RandomForests.pdf