CSE/ECE 343/543: Machine Learning
Assignment-2 Naive Bayes and Decision Trees
Max Marks: 100 (Programming:80, Theory:20) **Due Date**: 10/10/2020, 11:59PM

Instructions

- Keep collaborations at high level discussions. Copying/Plagiarism will be dealt with strictly.
- Late submission penalty: As per course policy.
- Your submission should be a single zip file **2018xxx_HW2.zip** (Where *2018xxx* is your roll number).
- Include only the **relevant files** arranged with proper names. A **.pdf report** explaining your codes with relevant graphs and visualization and theory questions.
- Do **NOT** include data files in your submission. It makes your files unnecessarily big while downloading.
- Ensure that everything required for a particular question is present in their respective files in terms of functions (not comments). Failure to do so would result in a penalty. Follow the following file structure for submission:

  2018xxx_HW1
  |– Q1.py
  |– Q2.py
  |– Q3.py
  |– Q4.py
  |– Report.pdf
  |– Dataset(folder)
  |– Weights(folder)
  |– Plots(folder)

- Remember to **turn in** after uploading on google classroom.
- Resolve all your doubts from TA's in their office hours **two days before the deadline.**
- **Document** your code. Lack of comments and documentation or improper file names would result in loss of 20% of the *obtained* score.

You are given three *small* datasets:

- **(Dataset A)** A dataset with each datum being a 784-dimensional vector and each datum has one of 10 labels (in 0-9)
- **(Dataset B)** A dataset, with each datum being a 2048 dimensional vector, and each datum having one of 2 labels (0 or 1)

- **(Dataset C)** Height-Weight dataset, with each datum being a 2 dimensional vector, and each datum having one of 2 labels (Male or Female)

1. (1.5+1.5+1.5+2.5+2.5+2.5+3 points) **t-SNE, PCA & SVD**. Use Dataset **A** and perform the following operations. You can use *sklearn* library for these operations.
   (a) Read about PCA and write a note on how it works.
   (b) Read about SVD and write a note on how it works.
   (c) Read about t-SNE and write a note on how it works.
   (d) Read about *stratified sampling* and perform a 80:20 stratified train-test split on the dataset. Comment on the class frequency of training and testing samples.
   (e) Use PCA on the dataset provided. Train a Logistic Regression model on the training set and report the test accuracy. Further, use t-SNE to analyze the training data.
   (f) Use SVD on the dataset provided. Train a Logistic Regression model on the training set and report the test accuracy. Further, use t-SNE to analyze the training data.
   (g) Compare the accuracy obtained while using PCA & SVD and write a note on the results obtained.

2. (3.5+1.5 points) (a) Use **dataset C** and train a Linear Regression model to predict weight based on the height of the person. Using bootstrapping, measure the bias & variance of the model and report them. *Hint - Refer to Lecture 6 (Revision).*
   (b) Assuming noise in the data to be zero, report the value of:
   $$MSE - Bias^2 - Variance$$
   and give a comment on the value obtained.

3. (7+3+6+3+16 points) Use *sklearn* Decision Tree(DT) and Gaussian Naive Bayes(GNB) classifier train it on **Dataset A & B** independently. Split the data into $60 - 20 - 20$ train-val-test split.
   (a) Find optimal depth as a parameter in-case of DT using Grid Search and use K-Fold cross validation to validate it. Implement your own K-Fold cross validation function from scratch and use for both GNB and DT. Make these function in such a way so that these can be used in future assignments.
   (b) For DT plot training and validation accuracy plot with respect to tree depth and write your analysis.
   (c) Plot validation v/s training accuracy plots for all the datasets on the optimal parameter that you choose from the previous part. Finally there should be 4 plots. (GNB=2, DT=2)
   (d) Save the best model, load the saved model to predict the results on the test data.
   (e) Write a function evaluation metric to evaluate testing data. Function should calculate accuracy, precision, recall, F1-Score, plot ROC-curve and return the confusion matrix. In case of multi-class data it should return Macro and Micro Average values. *Read Macro and Micro average values in Multi-class data.*

4. (25 points) Implement Gaussian Naive Bayes from scratch*(use of Numpy and Scipy allowed)* and compare results with Sklearn's implementation on dataset **A, B**

5. (2.5x4 points) (a) Given the dataset in table below for scheduling IPL matches. Using Decision Trees, predict PlayMatch using the other attributes. Show intermediate steps and trees while building the decision tree. Report the accuracy of the decision tree classifier.

   (b) Can you find some set of training examples that will get the algorithm to include the attribute Climate in the learned decision tree, even though the true target concept is independent of Climate? If yes, give the training set and the resultant tree. Otherwise, explain why it is not possible.

   (c) Divide the dataset in two training(D1 -D7) and test sets (D8-D14) . Build the decision tree using the training set and report the training and test accuracy. Explain why you think these are the results.

   (d) In this case, and others, there are only a few labelled examples available for training (that is, no additional data is available for testing or validation). Suggest a concrete pruning strategy, that can be readily embedded in the algorithm, to avoid overfitting. Explain why you think this strategy should work. (Note: Use the ID3 algorithm)

Day	Outlook	Climate	Humidity	Wind	PlayMatch
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

6. (3 points) You and your TA are using only two word vocabulary during an online office hours. Because of internet issue you missed one of word that your TA said: (w1 = Tough, w2 = course , w3 =???, w4 =course) Assume that your TA was generating words from this first-order Markov model:

   $p(tough|tough) = 0.7$

   $p(course|tough) = 0.3$

   $p(tough|course) = 0.5$

   $p(course|course) = 0.5$

   Given these parameters, what is the posterior probability of whether the missing word is "tough" or "course"?

7. (0.5+0.5+2+2+2 points) (a) What is the biggest advantage of decision trees when compared to logistic regression classifiers?

   (b) What is the biggest weakness of decision trees compared to logistic regression classifiers?

   (c) For the next problem consider n two dimensional vectors (x = x1,x2) that can be classified using a regression classifier. That is, there exists a w such that

$$y = \begin{cases} +1 & w^T x + b > 0 \\ -1 & w^T x + b = 0 \end{cases}$$

   (d) Can a decision correctly classify these vectors? If so, what is an upper bound on the depth of the corresponding decision tree (as tight as possible)? If not, why not?

   (e) Now assume that these n inputs are not linearly separable (that is, no w exists for correctly classifying all inputs using linear regression classifier). Can a decision tree correctly classify these vectors? If so, what is an upper bound on the depth of the corresponding decision tree (as tight as possible)? If not, why not?

8. (5 bonus points) Consider the Naıve Bayes model where Y is X is a vector of n Boolean variables.

$$X =< X_1, X_2..., X_n >$$

   Assume that $P(Y = 1) = \pi$.

   Show that $P(Y = 1|X)$ takes the form of a logistic function i.e.

$$P(Y = 1|X) = \frac{1}{(1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i))}$$

   Express the parameter of the logistic function θ in terms of the parameters of the Naıve bayes model.

   Hint - Since the X_i are Boolean variables, you need only one parameter to define $P(X_i|Y = y_k)$, where $y_k = 0, 1$. Define $\theta_{i1} \equiv P(X_i = 1|Y = 1)$, in which case $P(X_i = 0|Y = 1) = (1 - \theta_{i1})$. Similarly, use θ_{i0} to denote $P(X_i = 1|Y = 0)$.