CS578	**Fall 2021**
HW 1	
Handed Out: September 17, 2021	*Due: October 2, 2021*

Instructions for submission

1. Conceptual Part: **Submit a single PDF on Gradescope with all your answers. You should have been added, but if not the Entry Code is N8NWDK. Make sure you select the page corresponding to the beginning of each answer, else points might be deducted.** Your homework must be typed and must contain your name and Purdue ID. If you are using a late day, please indicate it at the top of the document. Do not send the instructors emails aboout Late Days.
2. Coding part: To submit your assignment, log into `data.cs.purdue.edu` (physically go to the lab or use ssh remotely) and follow these steps:
   (a) Place all of your code (`Check.py`, `Solution.py`, and any other files you used) in a folder titled `cs578-hw1`.**Make sure your folder is called this!**
   (b) Change directory to outside of `cs578-hw1/` folder (run `cd ..` from inside `cs578-hw1/` folder)
   (c) Execute the following command to turnin your code: `turnin -c cs578 -p hw1Fall2021 cs578-hw1`
   (d) To overwrite an old submission, simply execute this command again.
   (e) To verify the contents of your submission, execute this command: `turnin -v -c cs578 -p hw1Fall2021`.
   Do not forget the `-v` option, else your submission will be overwritten with an empty submission.

1 Review Questions

The review questions will not be graded, however you SHOULD submit your answers

1. Assume the probability of getting *head* when tossing a coin is λ.
   - What is the probability of getting the first head at the (k+1)-th toss?
   - What is the expected number of tosses needed to get the first head?
2. Let $f(x, y) = 3x^2 + y^2 - xy - 11x$
   - What is the partial derivative of f with respect to x ($\frac{\partial f}{\partial x}$)? Find $\frac{\partial f}{\partial y}$ as well.
   - Find a point (x, y) that minimizes f.

3.
   - Assume that $w \in \mathbb{R}^n$ and b is a scalar. A hyperplane in $\mathbb{R}^n$ is the set $\{x : x \in \mathbb{R}^n, w^T x + b = 0\}$. For n=2 and n=3, draw on paper an example of a hyperplane.
   - Assume we have two parallel hyperplanes: $\{x : x \in \mathbb{R}^n, w^T x + b_1 = 0\}$ and $\{x : x \in \mathbb{R}^n, w^T x + b_2 = 0\}$. What is the distance between these two hyperplanes?

2 Basic Concepts

1. Define in one sentence: (1) training set, (2) test set, (3) validation set. Your definition should use the notation described in class.
2. Can you use the validation set as a test set?
3. Define in one sentence: overfitting
4. True or False (and why): A learned hypothesis f has a training error e_{tr} and a testing error e_{ts}, where $e_{tr} > e_{ts}$. (1) can we say that f overfits to the training data? (2) Now, assume that $e_{tr} < e_{ts}$, does f overfit to the training data?

3 Decision Trees

1. The "Thrill and Romance" bookstore is interested in restocking its bookshelves based on their book sales data (summarized in the table below) by using a decision tree classifier. Each book is described using the number of its pages (an Integer), the author's reputation (Famous or Not), the book category (Detective, Romance or Tourism) and the color of the cover (Blue or Red).
   - What is the entropy of the target variable? (Buy)
   - What are the attributes considered by the algorithm? (hint: See lecture slides to see how continuous variables are treated)
   - What is the first attribute that the algorithm will split the data on? What is its information gain?
   - Due to a computer error some of the training examples attributes were deleted! Revise the decision tree training algorithm to deal with missing values in the training data.

Buy	Pages	Famous Author	Category	Cover Color
Y	300	Y	Detective	Blue
Y	50	N	Detective	Blue
N	100	Y	Romance	Blue
Y	150	Y	Romance	Blue
N	1000	N	Romance	Red
N	200	N	Detective	Blue
N	45	Y	Romance	Blue
Y	120	N	Romance	Blue
Y	350	Y	Romance	Blue
Y	142	Y	Detective	Blue
Y	72	Y	Detective	Red

2. Decision Tree Implementation: In the final part of this assignment you will have to implement the decision tree algorithm in Python. We will supply a template code for you to complete and the data.

   Prediction task In this assignment we will look into the Credit Approval problem, which contains attributes describing credit applications and a binary label describing the result. The data contains both binary, categorical and continuos values. **Note that some attribute values might be missing!**

   (a) You will need to implement two functions in the *Solution.py* file. The function *DecisionTree()* should contain your implementation of the decision tree algorithm, with your solution to the missing values problem. The function *DecisionTreeBounded(maxDepth)*, should also include a hyper parameter (maxDepth) which determines the maximum possible depth of the learned tree.

   You need to submit your codes to turnin. Please look at the instructions at the top of the handout for what needs to be submitted and the format. Your code should be runnable as:

   ```
   python3.6 Check.py
   ```

   This will load the training data (train.txt), validation data (validation.txt), and test data (test.txt) from the current directory and build a decision tree. It then evaluates it on both the training and test set, and prints the results. **Make sure you print the results exactly in the same format as they currently are in Check.py, else you will loose points. The code must complete running in less than 10 minutes, so you can just set the maxDepth value manually in submission.**.

   For example (the accuracy numbers are just random):

   ```
   python3.6 Check.py

   Training Result!
   ***************
   ```

```
Accuracy: 0.5
***************

Testing Result!
***************
Accuracy: 0.8
***************
```

(b) The best assignment to maxDepth (maximal depth of the learned decision tree) is not known in advance and needs to be tuned according to the available data. Since we are not allowed to use the test data when training the system, we set aside some of the training data, called a validation set, and use it to tune the hyperparamters of the learning algorithm.

You should follow these step: (1) use the training data for learning a decision tree, run the algorithm multiple times setting the value of maxDepth to different values (you can try different values, from maximal depth of one, and up until the maximal depth). Choose the best assignment to maxDepth by evaluating the learned tree on the validation set (**DO NOT use the test data directly!**) and report the result on test data.

In addition to the code implementation you should describe the results of your decision trees on the validation and testing data, based on different assignments to maxDepth. For this, please make two graphs plotting maxDepth (x-axis) vs. accuracy (y-axis) (one for validation set, one for test set) and analyze it in no more than 3 sentences. Make sure to discuss the trends in the graph and explain what value of maxDepth you decided. This part should be done in your Gradescope submission.

When submitting your code, it should only build the tree with the best maxDepth value.