DS 4400: Machine Learning and Data Mining 1 **(Due:** Nov 21

Homework Assignment #7

Professor: Miguel Fuentes-Cabrera *Name:* _____

Instructions: Please include the following information on the first page of your completed homework write-up:

1. Your name
2. DS 4400
3. Homework #7

You will submit **one files** to Gradescope for this homework:

- A .py or .ipynb file with all code used

Answers that are not supported by reasoning/work will not receive full credit. **Homework is due by 11:59 pm, via Gradescope, on the date above.** Late submissions will **not** be accepted, but you may receive extra credit for early submission (see syllabus for details).

You will also be graded on organization/neatness of the submitted files.

SVM (50 points)

(1) On Canvas, under Files-neeed-for-homeworks, is the **evs subset.csv** file, which contains a subset of the Electric Vehicles data set from the previous homework. Each row is a single car. The data consist of 91 electric vehicles and their measurements. Our goal is to predict the $y =$ drive ($-1 =$ Front, $1 =$ Rear) of the vehicle based on 2 other features:

- Top Speed (in MPH)
- Total Torque (in newtons by meters)

(a) Pre-process the data to prepare it for analysis by:

- Scaling the data (you may use whichever scaler you prefer).
- Separate the y feature into it's own NumPy array of appropriate -1's and 1's.
- Put the **two** x features of interest into their own array. Do not add an intercept column (Remember that in SVM the intercept is found separately).
- Split the data into training and test sets (we will skip validation).

(b) Plot the data on a scatterplot of the two x features with points colored by drive type. Discuss in a few sentences why you believe SVM might be a useful method for this problem, and how accurate you expect the model to be.

(c) Perform gradient descent with hard margin SVM on the training data set, and then use the model weights predict the observations in the test set. Report the classification accuracy of the test data.

Trees (50 points)

(2) On canvas, under Files-neeed-for-homeworks, is the **rmp df.csv** file, which contains 1589 reviews of professors from RateMyProfessor.com, and 24 features measured from those reviews. The target feature, and our goal, is to predict if a professor would be rated as "Would Take Again" (the first column in the data) using the other 23 features. One of those features is a likert scale (1-5, the review of the course difficulty) and the rest are all indicator variables such as if the professor requires attendance, or if they are "hilarious".

(a) In the data, 1183 out of the 1589 reviews rate the professor as "Would Take Again" ($Y = 1$). Calculate the entropy $H(Y)$ of the target feature.

(b) Given the two joint distributions below for "Would Take Again" (Y) and if the professor is "Respected" (X_1) or "Hilarious" (X_2), calculate the conditional entropy of both $H(Y|X_1)$ and $H(Y|X_2)$, then determine which X feature provides more information about the target feature. (**Note:** I have left the tables in terms of raw counts because I'm lazy; you will want to divide them all by 1589 to get the probabilities).

	$Y = 0$	$Y = 1$
$X_1 = 0$	188	246
$X_1 = 1$	218	937

	$Y = 0$	$Y = 1$
$X_2 = 0$	215	442
$X_2 = 1$	191	741

(c) To proceed with the analysis:

1. Separate the dataset into training, validation, and test sets.
2. Fit single decision trees of max depths 2, 3, and 4 using the training set. Use scikit-learn's **DecisionTreeClassifier** function, making sure that the splitting criterion is **entropy**.
3. Evaluate the performance of the three decision trees on the validation set. Report the precision, recall, accuracy, and F1 score of the three trees on the validation set.
4. Choose the max depth that seems to work best and use that model to predict the observations in the test set. Report the final test set accuracy and print out the final decision tree diagram fit to the **full data set**.
5. Print or plot out the feature importances for the final decision tree diagram fit to the **full data set**.

(d) Discuss in a few sentences if the single decision tree you fit in (c) seems to be a good model, what the feature importances tell you and if you think those values are meaningful, and if you think the algorithm could be improved (and if so, how). In the next part you will be fitting a random forest of trees using the max depth you identified for the single tree. What is the benefit of this?