**Homework 11 - Data Competition: Spam Detection**

Due December 3rd, 11:59 PM

IST 5520, Fall 2021, Chen

***Instruction***:

The objective of this data competition is for students to apply various predictive modeling techniques to accurately predict whether an email is spam or not. Students work in a pair of two students to finish the data competition. If you want to find a partner, you can submit a post on our discussion board. Students are also allowed to finish the work individually. Please work as early as possible so that you have sufficient tome to refine your final model by the due time.

Student pairs should work individually to finish the data competition. Any form of collaboration or help from other pairs/persons/parties is NOT allowed. Please submit your work by the due time.

***Data:***

The “train\_data.csv” contains a dataset of 3320 emails that is used to train classification models. Each row is an identical email. There are 58 columns explained below.

* 54 Columns “word\_freq\_make” through “char\_freq\_#”: Relative frequency of words/characters in the email;
* “capital\_run\_length\_average”: average length of uninterrupted sequences of capital letters;
* “capital\_run\_length\_longest”: length of longest uninterrupted sequence of capital letters;
* “capital\_run\_length\_total”: total number of capital letters in the e-mail;
* “class”: a number denotes whether the email is spam or not (1=spam, 0=not spam).

The “test\_data.csv” contains a dataset of 1381 emails that is used to evaluate the performance of classification models. The test dataset has the same data structure as the training dataset.

***Task:***

Apply various predictive modeling techniques to build a classification model that performs very well for predicting if a customer subscribe the product. Below is a list of possible techniques that could help you tune your models:

1. Deal with imbalanced dataset

For more details, refer to <https://imbalanced-learn.org/stable/introduction.html>.

1. Assign class weights
2. Use regularization parameters such as C in SVMs, and alpha in neural networks. For the effect of regularization parameter on model performance, refer to an example shown at <https://scikit-learn.org/stable/auto_examples/neural_networks/plot_mlp_alpha.html>.
3. Dimension reduction or feature selection
4. Tune hyperparamters
5. Use advanced models such as ensemble methods and deep learning

For more details, refer to <https://scikit-learn.org/stable/supervised_learning.html>.

1. Other methods……

You can only use the training dataset to train the classification models without touching the test dataset. Using test dataset to as a validation set to refine your models is also not allowed. You can normalize/transform the variables in the test dataset, but any resampling applied to the test dataset is not allowed. The test dataset is only used to evaluate the performance of your model. Your models need to be evaluated by AUC score [area under the ROC curve, using the method sklearn.metrics.roc\_auc\_score() to calculate].

Make sure that your jupyter notebook can be reproducible. Re-running the jupyter notebook will lead to the same results.

***Submit the Following Documents:***

1. Submit your jupyter notebook with detailed explanations. Your jupyter notebook should at least include the following contents:

* Specify your pair information in the jupyter notebook. Each pair only needs one submission.
* Specify contribution by each member. Students without significant contribution to the work will not receive the full regular points and extra credit.
* Data transformation and/or preprocessing.
* The final model trained from the training dataset.

***Note:*** I know you need to try multiple models and try many parameters. Please only submit the final model with the best parameters you find from your parameter tuning. Keep those time-consuming model optimization process in another jupyter notebook and submit it to the homework as a supplemental document. This ensures that your jupyter notebook with the final model can be re-run in a short time period.

* Test the performance of your final model using the test dataset. At least AUC score needs to be reported.

***Note:*** The jupyter notebook should display the results in consecutive sequence starting from 1. Make sure you click the “Kernel -> Restart & Run All” menu to re-run your jupyter notebook after you finish your code. A jupyter notebook that does not show the results in consecutive sequence will receive zero score.

1. Submit an HTML report generated from your jupyter notebook for your final model. On jupyter notebook, click “File -> Download as -> HTML” to generate the HTML report.
2. If you have used hyperparameter tuning, please submit your jupyter notebook for the hyperparameter tuning as a supplemental document.
3. Submit a one-page Microsoft Word document that explains the logic of your predictive modeling.

***Grading:***

***Regular Points***

All students’ works will be graded based on the AUC score of the final model. Correct work submitted by the due time will earn 40 points as a regular homework. Errors in the submission will reduce the points.

***Extra Credit***

Students who finish the work by the due time can earn extra credit according to the following rule:

* Submissions ranked as top 4 in terms of the AUC score of the final model will receive additional 2% extra credit for final grade;
* Submissions ranked 5rd to 8th best models in terms of the AUC score of the final model will receive additional 1% extra credit for final grade.