Stats 101C Kaggle Competition Final Project

Same requirements apply to both the Regression and Classification Projects

Grading of Final Project:

- 10% Competition performance
- 60% Report
- 30% Script verification

Competition performance grading:

The final project competition will be conducted on Kaggle. The competition is to create a supervised learning model that is able to make accurate predictions.

Students will be provided with a training data set and a test data set. The training data set contains the input variables X and the output values y.

The test data only contains the input variables X. Students will create predictions for the test data.

On Kaggle, there are two leaderboards: a public leaderboard and a private leader board. The public leaderboard uses a random selection of the test data for scoring. Students submit their predictions for the entire test data. Kaggle will display the performance on the public leaderboard (based on the public test set) while the competition is open. Results of the private leaderboard are hidden.

Once the competition ends, results of the private leaderboard will be revealed. Scoring for the Final Project Competition will be based on the results of the private leaderboard. It is very likely that the ranking of the private leaderboard will be different from the rankings of the public leaderboard.

Points as follows:

- Rank 1: 13 points out of 10
- Rank 2: 12 points out of 10
- 3: 11 points out of 10
- 4: 10 points out of 10
- 5: 9 points out of 10
- 6: 8 points out of 10
- 7: 7 points out of 10
- 8: 6 points out of 10
- 9: 5 points out of 10
- 10-12: 4 points out of 10
- 13-14: 3 points out of 10
- 15-16: 2 points out of 10
- 17-18: 1 point out of 10

Script verification

You will submit an R (or Python) script that produces the predictions you used for the competition.

The script should not produce any graphics or other output used in the report. The script should not produce predictions for models that were not used.

The script must use the trained model for making predictions, i.e. all predictions should be produced with the `predict()` function.

The script must not hard-code individual predictions manually.

The script should begin with loading the training and testing data sets as they appear on the Kaggle competition page.

The path for the csv files must point to the same folder that the script is located in. [Correct: read_csv("train.csv"), incorrect: read_csv("users/miles/desktop/stats/project/train.csv")]

The script must include lines to load any necessary libraries and any usage of random seeds so that the results can be reproduced on another computer. Any data manipulation, transformation, etc. must be included in the script.

The final line of the script should produce a csv file with the values that used for submission. There should be no additional editing of the csv file to upload to Kaggle.

The script will be run to verify that it does indeed produce the predictions submitted to Kaggle.

If the script fails to run or if the predictions you submitted to Kaggle do not match the output produced by your script, you will get a 0 for the script verification portion of your project grade. It is strongly recommended that you run the script on another computer to verify that it can indeed replicate the results submitted to Kaggle.

Report format and outline:

You will submit a PDF report explaining the model you fit.

The report must have clear headings separating sections.

Report must contain the following sections:

- Introduction: context and background info.
  - Cite any external sources
  - Can mention what variables you may believe to be associated with the response variable based on background information.
  - Approximately 100 words. Minimum length 80 words.
- Exploratory Data Analysis
  - Explore potential relationships between the variables.
  - Must include graphics showing relationships
  - Recommended: Transformations of some variables
  - Recommended: Converting some numeric variables into categorical variables
  - Recommended: Exploration of possible interactions between variables
  - Minimum 8 data visualizations. Maximum of 20. All graphs/visualizations must be accompanied by a description of its significance. Descriptions must be a minimum of 20 words, recommended something around 50 words.
- Preprocessing / Recipes
  - If you use recipes or perform preprocessing of variables, you must explain the steps you performed and the reasoning behind them.
  - Length will vary depending on preprocessing steps. ~ 100-500 words seems reasonable.
- Candidate models / Model evaluation / tuning
  - This section will discuss the various candidate models that were attempted.
  - Minimum of 5 candidate models. Maximum of 12 candidate models.
  - A brief description should accompany each candidate model.
  - Include a table listing of all candidate models attempted. Columns should include:

    - Model identifier
    - Type of model (e.g. linear regression, knn, random forest)
    - Engine
    - Recipe used or listing of variables in the model
    - Hyperparameters
  - Model evaluation and tuning
    - Discuss the evaluation and comparison of the candidate models that were attempted.
    - Students should use v-fold cross validation to measure the performance of the candidate models
    - Tuning of hyperparameters
    - Include a table summarizing the performance of each model. Columns should include
      - Model identifier
      - Metric score (most likely rmse)
      - SE of metric (if applicable)
    - Include a plot (like autoplot) comparing the performance of the different models.
- Discussion of final model
  - Discuss the selection of the final model used for generating predictions.
  - Discussion of strengths and weaknesses of the model.
  - Possible improvements, including what additional data could be useful
- Appendix: Final annotated script
  - The appendix of your report will be the final script used to produce results
  - Script must be in a monospace font (e.g. Courier, Monaco, Lucida Console, etc.)
  - The script must be annotated with comments
- Appendix: Team member contributions
  - List the contributions made by each team member