INFSCI 2595
Machine Learning

Fall 2021

Final Project Description and Instructions

The final project data are **graciously** provided by a local manufacturing company

- The data are real and are associated with a real research and development application which can benefit the company.

- **DISCLAIMER**: All variable names are given non-descript names and all ranges have been scaled to remove any reference to proprietary information from the company.
  - You are working with real data, but the origin of that data has been masked as requested by the company.

- **DISCLAIMER**: Because this is real data you **NOT** to upload the data to the internet. The data are **NOT** available on Github and should NOT be uploaded to any online repository/file share system.
  - The data can only be downloaded directly from Canvas and you should keep the data local to your computers.
  - If you have concerns about appropriate use of the data please contact Dr. Yurko.

Manufacturing and engineering companies rely on high fidelity physics computer simulations

- These simulations provide scientists and engineers the ability study complex physical phenomena.

- Trends can be analyzed, hypotheses can be tested, and experiments can be conducted **virtually** before they are attempted on real physical equipment.

- Computer simulations can therefore save large amounts of time and money due to preventing unnecessary physical experiments.

Computer simulations require specifying input variables and generate output variables

- The inputs to the computer simulation represent the operating characteristics of the materials and processes being simulated.
- The exact nature of the inputs depends on the application.
- The computer simulation models the complex physical phenomena associated with the process of interest.
- The computer simulation predicts the system behavior providing scientists and engineers with important outputs of interest.

Computer simulations require specifying input variables and generate output variables

- Consider the following example, I want to simulate the flight trajectory of a homerun hit by a baseball player.

- I need to specify the following inputs:
  - The velocity (speed and direction) of the ball “launched” from the bat.
  - The wind velocity (speed and direction).
  - The characteristics of the air (temperature and pressure).

- A computer simulation can model the air resistance (drag) force on the ball as it flies through the air. A user must specify how the drag is modeled and specify other important settings of the numerical solution scheme.

- Running the simulation for a given set of inputs will generate a flight trajectory of the ball.
  - A single output can be extracted from that trajectory to give the final distance traveled by the ball!

The data you are working with in the final project come from high fidelity physics simulations of an important manufacturing process

- We cannot discuss what that process is, but the ideas for how the computer simulation works is the same as the simple baseball homerun example.
- Inputs must be specified, the computer simulation is executed, and outputs are extracted.
- The data set you are working with consists of many inputs provided to the computer simulation and two important outputs predicted by the simulation.

Where do we, INFSCI 2595, fit in?

- Ideally, high fidelity computer simulations could be used to solve any problem!

- However, in order for the simulation to be as accurate as possible, the simulation can take a long time to run!

- Thus, the computer simulation cannot be used to give a “real time” prediction. It cannot make a prediction in a fraction of a second.

Where do we, INFSCI 2595, fit in?

- Machine learning methods are a useful tool to make the most out of the computer simulations.

- An experimental design strategy can be used to generate training data from a long running computer simulation.

- Machine learning methods can be used to model the output-to-input relationships.

- The machine learning methods therefore approximate the complex physical phenomena represented by the high fidelity computer simulations!

- Using machine learning methods for such applications is commonly referred as to creating surrogate models, reduced-order models, metal-models, or emulators.
  - My PhD thesis was focused on creating fully Bayesian emulators of long running physical simulations. This is how I initially got interested in machine learning!

Where do we, INFSCI 2595, fit in?

- Once machine learning models are trained, they can be used in place of the computer simulations.

- The machine learning models can be 100s if not 1000s of times faster than running the original computer simulations directly.

- The machine learning models therefore allow the computer simulations to be run effectively in real time!

Data description

- You are given 4 data sets in this project.

- The first is a small data set to get you started. You will work with the small data set first to get exposure to the goals of the project and working with the data.

- The remaining three data sets are associated with the “large” or “complete” problem.

Data description – starting small data set

- The starting data are in the CSV file: `small_train_data.csv`

- This data set consists of 6 variables:
  - 5 continuous inputs: `x07`, `x09`, `x10`, `x11`, `x21`
  - 1 continuous output: `response`

- The input names are consistent with the naming convention of the larger data set.

Data description – large or complete data

- Two sets of inputs are provided.

- The “x-variables” are in the CSV file: `train_input_set_x.csv`
  - 1 key/index ID column: `run_id`
  - 43 continuous inputs: `x01:x43`

- The “v-variables” are in the CSV file: `train_input_set_v.csv`
  - 1 key/index ID column: `run_id`
  - 41 continuous inputs: `v01:v41`

Data description – large or complete data

- One output data set is provided.

- The outputs are in the CSV file: `train_outputs.csv`
  - 1 key/index ID column: `run_id`
  - 1 continuous output: `response`
  - 1 binary outcome: `outcome`

- The `run_id` column is the “key” to join or merge the output data with the input data.

Project goals

- The continuous output, `response`, and the binary output, `outcome`, are both important quantities for the physical process.

- The binary output, `outcome`, has two levels or categories.
  - `outcome` equals either `event` or `non_event`.
  - You will train classifiers to classify event.

- Ideally, scientists and engineers want to identify input combinations that **<u>MINIMIZE</u>** the value of the continuous output, `response`, and **<u>MINIMIZE</u>** the probability that `outcome = event`.

Your primary goal of the project is therefore a multi-objective optimization problem!

- You must train regression models to predict the continuous output.
- You must train classification models to predict the binary output.
- You must identify the best models for each output.
- You must use your chosen best models to identify the input values that:
  - **MINIMIZE** the continuous output, `response`.
  - **MINIMIZE** the probability that the binary output, `outcome`, equals the `event`.
- Although this is truly a multi-objective problem, you will treat the optimization of each output separately.
  - You will compare the identified optimal input values based on minimizing the `response` vs minimizing the probability of `outcome = event`.

Your primary goal of the project is therefore a multi-objective optimization problem!

- IMPORTANT: You are not required to perform a formal optimization of the inputs.

- You will investigate performance through visualization.

- You will predict the `response` and predict the `outcome=event` probability with respect to the most important inputs and visualize those trends.

- Slides later in this presentation provide more details on how you should conduct such predictions.

Secondary project goals – feature engineering

- The inputs to the computer simulation can be defined several ways.

- Different definitions are used for different situations and have different interpretations.

- For example, if we have two variables a and b, we could instead define the two variables as a and the ratio of a and b, $r = a/b$.
  - Either approach works and is correct.
  - The variable definitions are however different.

Secondary project goal – feature engineering

- The “x-variable” inputs and “v-variable” inputs are two different ways of defining the inputs to the computer simulation.

- They are related, similar to the simple example on the previous slide, but the “v-variables” are not ratios of the “x-variables”.

- The scientists and engineers are interested to know if the input definition impacts the performance of the machine learning models.

Secondary project goal – feature engineering

- Therefore, you will train regression models using the “x-variables” and compare their performance to the same regression models using the “v-variables”.

- You will also train classification models using the “x-variables” and compare their performance to the same classification models using the “v-variables”.

- You will compare the performances of all models using the two different input definitions.
  - You must identify if one definition leads to better performing models!
  - If the input definitions do not matter, then that is a useful conclusion as well!
  - You must make that comparison for the regression and classification problems.

Getting started – begin simple!

- This is a challenging project.
- It is high dimensional (many inputs) and you must ultimately interpret the trends of two different outputs with respect to the inputs.
- Rather than jumping in to the complete problem right away, instead you will start with a simplified problem.
- You will work through most of the major steps of the project for this simplified problem using linear model techniques to help you become familiar with the data and some of the output-to-input trends for the regression problem.

Getting started – fitting models

- The simplified problem comes from an experimental design that ran the computer simulation focusing on the behavior of just 5 of the “x-variable” inputs.
    - Thus, you will focus on just the “x-variables” at the start.
- You will fit non-Bayesian and Bayesian linear models to predict the continuous output, response, with respect to the 5 inputs.
- You will consider additive features, interactions, and various basis functions.

Getting started – visualizing predictive trends

- You must identify the best models from the candidate set of models that you tried.

- You must identify which inputs you wish to visualize trends with respect to.

- You must make predictions of the trend, including the confidence interval, and prediction interval of the continuous output, `response`, with respect to the inputs.

- From your visualizations, can you identify the values of the 5 inputs that minimize the `response`?

Complete or large problem

- After you completed the “getting started” simplified problem, you will move to the larger problem.

- You will train regression models to predict the continuous output, `response`, then you will train classification models to predict the categorical output, `outcome`.

- You will train models using the “x-variable” inputs and you will train models using the “v-variable” inputs.
    - The two input sets should NOT be mixed. They should be kept separate from each other.

- You will use non-Bayesian resampling techniques to train and assess the model performance.

The project is open ended

- No template is provided.

- An Rmarkdown is provided to give an example of reading in the data.
  - It also shows how to join the 3 larger data sets appropriately.
  - It also shows how to save a model object and load that model in again.

- Specific requirements are listed next, and those requirements can help guide you through the predictive modeling application.

Project consists of 5 areas

Part i: Exploration

- It is always important to explore and study your data before starting a modeling exercise.

Part ii: Getting started: small, simplified problem – linear models

- Fit linear models to predict the continuous output as a function of the 5 inputs in the small simplified design.
- You will use non-Bayesian and Bayesian approaches.

Part iii: Complete problem: regression – linear and non-linear methods

- Train regression models to predict the continuous output as a function of the 43 “x-variables”.
- Train regression models to predict the continuous output as a function of the 41 “v-variables”.
- You will use resampling to train, tune, and assess performance of multiple models, including 2 methods not explicitly discussed in lecture.

Part iv: Complete problem: classification

- Train binary classifiers to classify outcome=event as a function of the 43 “x-variables”.
- Train binary classifiers to classify outcome=event as a function of the 41 “v-variables”.
- You will use resampling to train, tune, and assess performance of linear and non-linear methods, including 2 methods not explicitly discussed in lecture.

Part v: Interpretation and “optimization”

- Use the best models to identify the most important variables that influence continuous output and the probability of the event.
- Does the input set influence model performance? Essentially, are models better when the “v-variables” are used as inputs instead of the “x-variables”?
- Visualize the behavior of the continuous output with respect to the most important inputs.
- Visualize the behavior of the event probability with respect to the most important inputs.
- Recommend input settings to use to minimize the value of the continuous output AND recommend input settings that minimize the event probability.

Part i: Exploration

- Visualize the distribution of the variables in the data set.
  - Distributions of the inputs – the “x-variables” and “v-variables”.
  - Distribution of the continuous output and the counts of the binary output.
  - Compare the distributions of the 5 inputs in the “Getting started” simplified design to their distributions in the complete design (the input names are consistent between the small and large data sets).
  - Compare the distribution of the continuous output in the “Getting started” simplified design to its distribution in the complete design.

- Consider breaking up the continuous variables based on the binary output.
  - Are there differences in input values based on the binary output levels?
  - Are there differences in the continuous output based on the binary output levels?

- Visualize the relationships between the “x-variable” inputs, are they correlated? Visualize the relationships between the “v-variable” inputs, are they correlated?

- Visualize the relationships between the continuous output and the “x-variable” inputs and the “v-variable” inputs.
  - Compare the continuous output relationships to the 5 inputs in the “Getting started” simplified design with the relationships in the complete larger design. Are they similar?

- How can you visualize the behavior of the binary outcome with respect to the inputs?

Part ii: Small problem linear models - iiA)

- Use `lm()` to fit linear models with the small data set. Try the following:
    - All 5 inputs with additive features.
    - All pair-wise interactions with the 5 inputs.
    - Polynomial basis of your choice applied to all 5 inputs.
    - 3 additional basis functions of your choice.
        - For example, one input interacts with a polynomial basis of another input, or spline basis applied to separate inputs, etc…

- Which of the 6 models is the best? What performance metric did you use to make your selection?

- Visualize the coefficient summaries for your best two models. How do they compare?

Part ii: Small problem linear models - iiB)

- Use Bayesian linear models to fit 2 of the models you fit with `lm()`.
- You may use the Laplace Approximation approach we have used in lecture and the homework assignments.
- Or, you may use `rstanarm`'s `stan_lm()` function to fit full Bayesian linear models with syntax similar to R's `lm()` function.
    - Getting started with rstanarm
    - Using the stan_lm() function
- Which model is the best? What performance metric did you use to make your selection?
    - Visualize the posterior distributions on the coefficients for your best model.
- For your best model: study the uncertainty in the noise (residual error), σ. How does the `lm()` maximum likelihood estimate (MLE) on σ relate to the posterior uncertainty on σ?

Part ii: Small problem linear models - iiC)

- You must make predictions with the top 2 models in order to visualize the trends of the continuous output with respect to the inputs.

- You may use non-Bayesian or Bayesian models for the predictions.

- You must decide which inputs you wish to visualize the trends with respect to.
  - The primary input should be used as the x-aesthetic in a graphic.
  - The secondary input should be used as a facet variable, use 4 to 6 unique values of the secondary input (creating 4 to 6 facets).
  - You must decide what values to use for the remaining inputs.

- Whether you use non-Bayesian or Bayesian models, **you MUST include the predictive mean trend, the confidence interval on the mean, and the prediction interval** of the continuous output.

- Are the predictive trends different between the top 2 models you selected?

- Which input values correspond to minimizing the continuous output? Are you confident that the value will be minimized?

Part iii: Complete problem regression – iiiA)

- You must train, evaluate, tune, and compare more complex methods via resampling.
  - You may use `caret` or `tidymodels` to handle the training, testing, and evaluation.
- You must train and tune the following models:
  - Linear models:
    - Linear additive features
    - All pair-wise interactions between the inputs
    - One model with features consistent with the features from the top performing models in the simplified “Getting started” portion of the project.
  - Regularized regression with Elastic net
    - All pair-wise interactions between the inputs
    - A complex model with interactions and features consistent with the features from the top performing models in the simplified “Getting started” portion of the project
  - Neural network
  - Random forest
  - Gradient boosted tree
  - 2 methods of your choice that we did not explicitly discuss in lecture.

For iiiA) you must predict the continuous output with respect to the “x-variables”

Part iii: Complete problem regression – iiiB)

- You must train, evaluate, tune, and compare more complex methods via resampling.
  - You may use `caret` or `tidymodels` to handle the training, testing, and evaluation.
- You must train and tune the following models:
  - Linear models:
    - Linear additive features
    - All pair-wise interactions between the inputs.
  - Regularized regression with Elastic net
    - All pair-wise interactions between the inputs
    - A complex model with interactions and basis features of your choice.
  - Neural network
  - Random forest
  - Gradient boosted tree
  - 2 methods of your choice that we did not explicitly discuss in lecture.

For iiiB) you must predict the continuous output with respect to the “v-variables”

Part iii: Complete problem regression – iiiA) and iiiB)

- You must decide the resampling scheme, what kind of preprocessing options you should consider, and the performance metric you will focus on.

- You must apply the same resampling scheme to the “x-variable” based models and the “v-variable” based models.

- You must identify the best model.

Part iv: Complete problem classification – ivA)

- You must train, evaluate, tune, and compare binary classifiers via resampling.
  - You may use `caret` or `tidymodels` to handle the training, testing, and evaluation.
- You must train and tune the following models:
  - Logistic regression:
    - Linear additive features
    - All pair-wise interactions between the inputs.
  - Regularized logistic regression with Elastic net
    - All pair-wise interactions between the inputs
    - A complex model with interactions and basis features of your choice.
  - Neural network
  - Random forest
  - Gradient boosted tree
  - 2 methods of your choice that we did not explicitly discuss in lecture.

For ivA) you must predict the continuous output with respect to the “x-variables”

Part iv: Complete problem classification – ivB)

- You must train, evaluate, tune, and compare binary classifiers via resampling.
  - You may use `caret` or `tidymodels` to handle the training, testing, and evaluation.
- You must train and tune the following models:
  - Logistic regression:
    - Linear additive features
    - All pair-wise interactions between the inputs.
  - Regularized logistic regression with Elastic net
    - All pair-wise interactions between the inputs
    - A complex model with interactions and basis features of your choice.
  - Neural network
  - Random forest
  - Gradient boosted tree
  - 2 methods of your choice that we did not explicitly discuss in lecture.

For ivB) you must predict the continuous output with respect to the “v-variables”

Part iv: Complete problem classification – ivA) and ivB)

- You must decide the resampling scheme, what kind of preprocessing options you should consider, and the performance metric you will focus on.

- You must apply the same resampling scheme to the “x-variable” based models and the “v-variable” based models.

- Which model is the best if you are interested in maximizing Accuracy compared to maximizing the area under the ROC curve?

Part v: Interpretation and “optimization”

- After you have selected the best performing models consider:
  - Does the model performance improve if the “v-variables” are used instead of the “x-variables”?

- Identify the most important variables associated with your best performing models.

- Visualize the predicted continuous output as a function of your identified most important variables.

- Visualize the predicted event probability as a function of your identified most important variables.

- Based on your visualizations, what input settings are associated with minimizing the continuous output?

- Based on your visualizations, what input settings are associated with minimizing the event probability?

- BONUS +10 points: Optimize the inputs using `optim()`.

Two additional methods

- You may use the same two methods for both the regression and classification portions of the project.
    - If however, you select a method that cannot be used for both regression and classification, then you will need to select an additional method.

- Potential methods to consider:
    - Support Vector Machines (SVM) – classification and regression
    - Naïve Bayes – classification
    - Generalized Additive Models (GAM) – classification and regression
    - Multivariate Additive Regression Splines (MARS) – classification and regression
    - Partial Least Squares (PLS) – classification and regression
    - Deep neural network – classification and regression
    - K-nearest neighbors – classification and regression
    - Stacked models

- Please see Ch 6 in the caret documentation for a complete list of all available methods in `caret`.
- Please see the tidymodels parsnip list of available models for more details.

Interpretation and visualization help

- Chapter 16 in the HOML provides useful discussion on interpretable machine learning.

- Provides code examples for visualizing model behavior and interpreting the graphics.

Homework assignments include examples working with `caret`

- You may use caret to perform all the resampling, tuning, and evaluation for the project
- However, you may use `tidymodels` instead of `caret`.
- `tidymodels` provides modeling aligned with the philosophy of the `tidyverse`, created by the developers of `caret`.
- If you are interested to learn `tidymodels` please see:
- https://www.tidymodels.org/
- Try out some of the “Get Started” tutorials.

Applied machine learning examples available on Canvas provide both `caret` and `tidymodels` examples

- Week 01 – Airfoil example problem
    - Example EDA, linear models, and regression models with `caret`

- Week 02 and Week 03 – examples
    - Regression application with `tidymodels` – Concrete data
    - Binary classification application with `tidymodels` – Ionosphere data

Bonus points – model tuning

- In addition to attempting formal optimization of the inputs, you may earn bonus if you attempt the following:

- Tune the machine learning methods with an approach other than grid search (we will use grid search in lecture and homework).
    - Up to BONUS +10 points for using an iterative/adaptive tuning strategy.

- Examples to get you started:
    - Bayesian optimization – tidymodels example here
    - Racing methods – tidymodels example here, Julia Silge blog post here
    - Adaptive resampling – caret documentation here

Bonus points – neural networks

- In lecture, we will use the `neuralnet` and `nnet` packages for training neural network models.

- However, Torch is available natively in R.

- Up to BONUS +10 points for training and tuning neural networks with Torch.

- Please see the following to get started:
  - RStudio AI blog announcement
  - torch CRAN page

Test set predictions

- A test set of just input values will be provided late November.
- You must predict the continuous response and the event probability using this test set.
- You will upload your predictions to a website. The website will provide the performance metrics associated with your predictions.
- More to come on this later!

Project submission

- You must submit the RMarkdown source .Rmd file and the associated rendered HTML document.

- It is recommended that you create separate RMarkdowns for the different portions of the project. This way you can work in a more modular fashion and will not have a single enormous file.

- **Project must be submitted no later than Friday December 10, 2021 at 11PM EST (Pittsburgh, PA local time).**