CS 1675
Introduction to Machine Learning

Fall 2021

Final Project Description and Instructions

Dr. Joseph P. Yurko (Pitt)

Anastasia Sosnovskikh (PPG, Pitt 2021)

Final project sponsored by PPG Industries

- Fortune 500 Company
- Global supplier of paints, coatings, and specialty materials
- Largest coatings company in the world by Revenue
- Headquarters in Pittsburgh, PA and operates in 70+ countries around the world
- Please see ppg.com to learn more!

Surface coatings are critically important

- Surface coatings play a role in the manufacturing of products from common household goods, eyeglasses, buildings, cars, planes, and more!

- Coatings prevent corrosion and prolong the useful life of industrial materials, components, and machinery.

- Without a properly designed surface coating, the materials we interact with would not last as long as we want them to!

Surface coating manufacturing

- A coating is created by combining multiple constituent materials together.
  - Think of the ingredients of a cooking recipe.

- The constituents are combined in a manufacturing process following a specific set of operating conditions.
  - Think of the steps or instructions of a cooking recipe.

Surface coating testing

- Each batch of material produced is tested by applying the coating to a specimen.

- The specimen is subjected to an *accelerated life test.*
  - The specimen is exposed to extreme temperatures and humidity for long periods of time.
  - The test conditions force corrosion to occur much faster than would normally occur to simulate years of typical material use.

- After the test is complete, the amount of corroded surface is recorded.

Surface coating design

- Properly designing coating materials requires experts in chemistry and chemical engineering, materials science, manufacturing, experimental design, materials testing, quality control, and more.

- Experiments are performed to find the optimal material chemistry and manufacturing process conditions that minimize the amount of corroded surface after a test.

Where do we, CS 1675, fit in?

- Recently, machine learning methods have become an important tool to aid traditional design techniques.

- Machine learning models are trained using historical data to predict material performance.

- Machine learning expertise is required to know how to properly apply the methods!

Data description

- You are given a data set consisting of inputs and an output of interest.
  - The data come from a mixture of previous experimental designs, as well as production data.

- The inputs consist of three groups of variables:
  - Chemistry variables: `x1`, `x2`, `x3`, `x4`
  - Manufacturing process variables: `v1`, `v2`, `v3`, `v4`, `v5`
  - Machine used to manufacture the coating: `m`

- The output is the fraction of the specimen surface that corroded after the test completed.
  - Response variable: `output`

Data description

- The data are stored in a table within a CSV file.
- One row corresponds to a test result.
- Each row consists of the 4 chemistry inputs, 5 process inputs, the machine input, and the response.
- Thus, you are provided with the inputs associated with the tested result!

Your primary goal is to optimize the coating!

- You must train models to predict the fraction of corroded surface per test (the `output`) as a function of the inputs.

- You must identify the best model and use that model to identify the input values that **minimize** the `output`.

- You are not required to perform a formal optimization of the inputs. You will investigate performance through visualization, as described later.

Secondary project goals - classification

- Although predicting the fraction of the corroded surface is important, some scientists and engineers are also interested in understanding which inputs influence the fraction achieving a threshold value.

- **They want to know which inputs are most important at causing the fraction of corroded surface to be less than 0.33.**

- You must train binary classifiers to classify if the surface will corrode less than 33%.
  - The surface corroding less than 33% is the EVENT, and the surface corroding greater than 33% is the NON-EVENT.

- You must select the best performing binary classifier and rank the input variables based on their importance at causing the EVENT to occur.

Secondary project goals – feature engineering

- Historically, scientists and engineers (the Subject Matter Experts or SMEs) have not directly used the provided inputs when trying to design the coating to minimize corrosion.

- Instead, they use features **derived** from some of the provided inputs.

- You must train models using the provided inputs **AND** models using a mixture of provided inputs and derived features to identify if the derived features are as important as the SMEs believe they are.

Inputs and derived features

- The chemistry inputs, `x1`, `x2`, `x3`, and `x4`, are **fractions** between 0 and 1.
- They provide the fraction of the coating material associated with each constituent, however their sum does NOT equal 1.
- A “balance” constituent, `x5`, is also present in the coating material. The fraction associated with the balance is calculated as:

```
x5 = 1 – (x1 + x2 + x3 + x4)
```

- Some of the SMEs feel the value of the “balance” constituent, `x5`, is important for predicting the `output`.

Inputs and derived features

- Some of the SMEs believe that ratios of the constituents are more important than the values of the individual constituents.

- The “w” ratio depends on `x2`, `x3`, and `x4` and is defined as:

```
w = x2 / (x3 + x4)
```

- The “z” ratio depends on all constituents and is defined as:

```
z = (x1 + x2) / (x4 + x5)
```

Inputs and derived features

- The manufacturing process inputs, `v1`, `v2`, `v3`, `v4`, and `v5`, represent how the chemical constituents are combined to create the coating.

- Some of the SMEs feel the product of v1 and v2 is important.

- Their product is defined as:

```
t=v1*v2
```

Secondary project goals – categorical input

- The machine used to manufacture the coating is provided by the `m` input variable.

- The `m` input is a categorical input. Each unique value (level or category) corresponds to a different machine.

- **The SMEs are interested to know if the machine influences the response.**

Response considerations

- Although the problem has been formulated as predicting the fraction of corroded surface, you should **NOT** predict the fraction directly.
- The `output` is a fraction bounded between 0 and 1.
- You should instead **transform** the output by applying a logit transformation and **your regression models should be trained to predict the logit-transformed response**.
- The logit-transformed response can be calculated in R as:

```
y = boot::logit(output)
```

The project therefore consists of regression and classification tasks

REGRESSION

- Predict the logit-transformed response, `y`, as a function of the provided inputs: `x1:x4`, `v1:v5`, `m`.
  - This approach is referred to as the "**base feature**" regression models.

- Predict the logit-transformed response, `y`, as a function of the derived features, `x5`, `w`, `z`, and `t`, and some of the provided inputs: `x1:x4`, `v1:v5`, `m`.
  - It is up to you whether you should use all of the provided inputs or not. Be careful though if you use ALL provided inputs and ALL derived features…
  - This approach is referred to as the "**expanded feature**" regression models.

The project therefore consists of regression and classification tasks

<u>CLASSIFICATION</u>

- Classify if the binary outcome is the EVENT as a function of the provided inputs: `x1:x4`, `v1:v5`, `m`.
  - This approach is referred to as the "**base feature**" classification models.

- Classify if the binary outcome is the EVENT as a function of the derived features, `x5`, `w`, `z`, and `t`, and some of the provided inputs: `x1:x4`, `v1:v5`, `m`.
  - It is up to you whether you should use all of the provided inputs or not. Be careful though if you use ALL provided inputs and ALL derived features…
  - This approach is referred to as the "**expanded feature**" classification models.

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 calculate the derived quantities, including those derived from the inputs, the logit-transformed response, and the binary outcome.
  - 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: Linear models

- Fit linear models to predict the logit-transformed response using the “base features” and the “expanded features”.
- You will use non-Bayesian and Bayesian approaches.

Part iii: Regression models: linear and non-linear methods

- Train regression models to predict the logit-transformed response using the “base features” and the “expanded features”.
- You will use resampling to train, tune, and assess performance of multiple models, including 2 methods not explicitly discussed in lecture.

Part iv: Binary classification

- Train binary classifiers to classify the EVENT using the “base features” and the “expanded features”.
- 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 the logit-transformed response and the probability of the EVENT.
- Visualize the behavior of the logit-transformed response with respect to the most important inputs.
- Visualize the behavior of the probability of the EVENT with respect to the most important inputs.
- Recommend input settings to use to minimize the fraction of corroded surface.

Part i: Exploration

- Visualize the distribution of the variables in the data set.
  - Distributions of the inputs – “base features” and the derived features.
  - Distribution of the `output` and the logit-transformed response.

- Consider breaking up the continuous variables based on the categorical variable `m`.
  - Are there differences in input values based on the discrete groups?
  - Are there differences in `output` based on the discrete groups?

- Visualize the relationships between the inputs (“base features” and derived features), are they correlated?

- Visualize the relationships between the responses (`output` and the logit-transformed response) with respect to the inputs (“base features” and derived features).

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

Part ii: Linear models - iiA)

- Before using more advanced methods, get a feel for the behavior of the logit-transformed response as a function of the inputs using linear modeling techniques.

- Use `lm()` to fit linear models. You must use the following:

- 3 Models using the "base feature" set:
  - All linear additive features
  - Interaction of the categorical input with all continuous inputs
  - All pair-wise interactions of the continuous inputs
- 3 Models using the "expanded feature" set:
  - Linear additive features
  - Interaction of the categorical input with continuous features
  - Pair-wise interactions between the continuous features
- 3 Models linear basis function models:
  - It is your choice if you want to only use the "base feature" set or include features from the "expanded feature" set.
  - It is your choice which types of basis functions to use and which input/feature the basis should be applied to.
  - Possible basis functions to consider: polynomials, splines, sin/cos
  - You may include as many interactions between inputs, features, and basis features, and the categorical input as you like.
  - If you do not want to use interactions, that's ok too! It's up to you!

Part ii: Linear models – iiA)

- You must therefore train 9 different models!
- Which of the 9 models is the best? What performance metric did you use to make your selection?
- Visualize the coefficient summaries for your top 3 models.
- How do the coefficient summaries compare between the top 3 models?

Part ii: 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: Linear models – iiC)

- You must make predictions with the top 2 models in order to visualize the trends of the logit-transformed response with respect to the inputs.
- You may use non-Bayesian or Bayesian models for the predictions.
- You must decide which inputs or features 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/features.
- 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** on the (logit-transformed) response.
- Are the predictive trends different between the top 2 models you selected?

Part iii: Regression models

- 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:
    - Additive features using the "base feature" set
    - Additive features using the "expanded feature" set
    - Your top ranked linear model from Part ii)
    - Another linear model of your choice from Part ii)
  - Regularized regression with Elastic net
    - Interact the categorical variable with all pair-wise interactions of the continuous features.
    - The most complex model you tried in Part ii)
  - Neural network
  - Random forest
  - Gradient boosted tree
  - 2 methods of your choice that we did not explicitly discuss in lecture.

The linear methods are included in Part iii) to give context to the performance of the advanced non-linear methods

Part iii: Regression models

- You must train and tune the neural network, random forest, and the gradient boosted tree with the "base feature" set and again with the "expanded feature" set.
- You may decide whether to use the "base feature" set or "expanded feature" set for the 2 additional models you select.
- You must decide the resampling scheme, what kind of preprocessing options you should consider, and the performance metric you will focus on.
- You must identify the best model.

Part iv: Binary classification

- 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:
    - Additive features using the "base feature" set
    - Additive features using the "expanded feature" set
    - Your top ranked linear model from Part ii)
    - Another linear model of your choice from Part ii)
  - Regularized logistic regression with Elastic net
    - Interact the categorical variable with all pair-wise interactions of the continuous features.
    - The most complex model you tried in Part ii)
  - Neural network
  - Random forest
  - Gradient boosted tree
  - 2 methods of your choice that we did not explicitly discuss in lecture.

Part iv: Binary classification

- You must train and tune the neural network, random forest, and the gradient boosted tree with the "base feature" set and again with the "expanded feature" set.

- You may decide whether to use the "base feature" set or "expanded feature" set for the 2 additional models you select.

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

- 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 when the derived features in the “expanded feature” set are included?

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

- Visualize the predicted logit-transformed response as a function of your identified most important variables.

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

- Based on your visualizations, what input settings are associated with minimizing the logit-transformed response?
  - Do the optimal input settings vary across the values of the categorical variable?

- BONUS +10 points: Optimize the inputs/features for 2 values of the categorical variable 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 logit-transformed response and the probability of the EVENT 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).**