**Stat. 652 Statistical Learning**

**The Five Steps** for implementing a Data Science Project or Machine Learning Project

**Step 1:** Determine the data to be used and where it will come from.

This is the first two parts of what is known as ETL (extract, transform, and load), the second part is step 2.

Some things to consider:

1. What columns do you want? Make a list of the input and output variables. There may be more than one output variable that is going to be useful, try to ask for it at the start of your ETL process.
2. What rows do you want? This is related to the time frame of the collected data. When do you want your data to start to be collected and when do you want it to end.

**Step 2:** Load the data into your software within a Project Folder. Clean the data.

This may be reading a .csv or .xlsx file. Or it might be to read in a file or files with a different file extension such .json or .xml (common file formats for data stored online). Or you might be a proprietary file format that you will need to have coverted into another format. Or you might be through a given API key to read data from a server on the internet.

Remove any ID variables.

Remove any other target variables that are directly related to the target variable you are using.

Remove any duplicate or very highly correlated columns.

Check for duplicate rows in your dataset. Sometime the same row is repeated. If there appears to be a copy of the same row in the dataset you should consider removing it, but only if you are certain it is a duplicate.

Use AutoEDA to explore your data. Determine the missing rates for all column, impute if possible.

Be aware that some variables may return with 100% missing values. Some may return with 90% missing. So many rows of your dataset will need to be dropped if you plan to drop rows with any missing values in a column. There is a balancing process that needs to be taken to drop columns with too high of a missing rate but to retain as many rows as possible you may choose to use an imputation method to fill in the missing data.

For categorical variables you should code them to be factors in R. When working with a categorical target variable you should note which level of the variable comes first, that will be the level predicted by the algorithm. If the variable is a Yes or No variable, because No comes first in the alphabet (or it is the most frequent level of the variable) it is the level of the variable that might be predicted. If you want to predict Yes, then you wil need to reorder the levels of the categorical target variable.

**Step 3:** Develop the models you plan to investigate. You will likely be iterating through many models using the same algorithm and then iterating though many different models to determine a collection of best models for a particular model and then selecting the best model overall.

Determine the Null Model y ~ 1

Start with the Full Model, y ~ .

**Models:**

* kNN, used with numeric features
* Naive Bayes, used with binary features
* Decision Trees:
  + Decision Tree using Divide-and-Conquer, for numeric features
  + Rules based Trees using Separate-and-Conquer, for nominal features
* Ensamble methods:
  + Boosting
  + Bagging
  + Random Forests, Boruta
* Linear Regression for Prediciton using Regualization
* Logistic Regression for Classification using Regularization
* SVMs
* Neural Networks

**Step 4:** Fit the models you plan to consider from Step 3.

Fit the models and remove any variables, either automatically using the model/algorithm you are fitting or drop any variables that are causing problems running the algorithm.

It is likely that after clearning your data you will try to start fitting a model and the model does not run the first time. You will have to debug the code and decide what further data cleaning steps are necessary to get the model to run.

**Models and debugging consideration:**

* kNN This classification algorithm uses distance so there can be no missing data. To start it is recommened that you start with **all numeric variables/columns** of data to work with.
* Naive Bayes For this classification algorithm is commonly used with **all categorical variables/columns** that are binary.
* Decision Trees For these models it is possible to use both numeric and categorical variables. Random Forests are one of the best tree based alagorithms to apply. Also, Boosting algorithms work very well. It is not recommended to build single trees because Decision Trees are “weak learners,” which means they produce different results each time the algorithm is run.
* Feed forward Neural Networks are the next step in developing machine learning modesl.

**Step 5:** Try to improved the models and determine what the final model is. Will be it one model or an ensamble of modles that averaged in some way? If you use Accuracy as the evaluation metric and a logistic regression is the best model, note in the end that you have fitted on model. If a Random Forest or Boosting Algorithm is selected, these are ensamble methods so the final model is determined from a large collection of models.

Tune your model. Try to determine the best values for the tunning parameter(s) in your final model. Even though the best model has been determined, it is worth trying to improve it futher by tuning the model paramter to make sure you are using the best value of the tuning parameter.

Determine how the final model will be served to your clients. The last step is to fit the data a final time. Do you use the model you have use to determine it is the best model or do you refit that model on the entire dataset, using both the training and testing data together? While this is done sometime, it is also reasonable to use the final model you have determined using your training or validation data.