Lab 1 – SIRCSS – Machine Learning for Social Science

Although not obligatory in any way, I recommend—for good practices and learning purposes—to create a document that includes code, relevant output, as well as brief answers to the questions. I recommend the use of RMarkdown for this purpose. But Word or any software like it is of course fine too.

Given the rather short amount of time available, you may likely not finish all tasks, and that is totally understandable and fine. I advise doing the tasks in order. Some tasks are marked as bonus tasks. At the end of the session, I will upload the solutions. Lastly, refer to the R helpfile for code that might be useful in solving the tasks; it includes code for solving similar types of problems using the R functions referenced in this document.

Quiz-warm up:

Here are four **optional** warm-up questions that help recap some central ideas from the lecture.

1. For each of the scenarios described below (a–c), indicate whether it is a *classification* or *regression* problem, and further whether we are most interested in *inference* or *prediction*:

   a. A medical researcher collects data on patients, including age, blood pressure, cholesterol levels, and whether they develop heart disease within five years (yes: 1, no: 0). The goal is to understand which factors influence the risk of heart disease.

   b. A bank wants to predict the amount of money (in dollars) that customers will spend on their credit cards next month, based on their spending history and demographic information. The primary goal is to forecast future spending accurately.

   c. A company is analyzing customer reviews to determine whether each review is positive or negative. The goal is to automatically classify new reviews as positive or negative to monitor customer satisfaction.

2. Figure 1 (see next page) displays a conceptual graph of the so-called *bias-variance trade-off.* Your task is to pair each shaded region (denoted *a–c*) with one of the fitted lines (denoted *i–iii*) found in the scatter-plots below the bias-variance plot. For example, one possible pairing-configuration could be $\{a : ii\}$, $\{b : i\}$, $\{c : iii\}$.

3. Which among the following are important aspects distinguishing *parametric* and *non-parametric* supervised learning models?

   a. Non-parametric methods make strong assumptions about the functional form of models; parametric methods do not.
   b. Parametric methods make strong assumptions about the functional form of models; non-parametric methods do not.
   c. Parametric methods are generally more appropriate for *inference* tasks compared to non-parametric methods.
   d. Non-parametric methods are generally more appropriate for *inference* tasks compared to parametric methods.

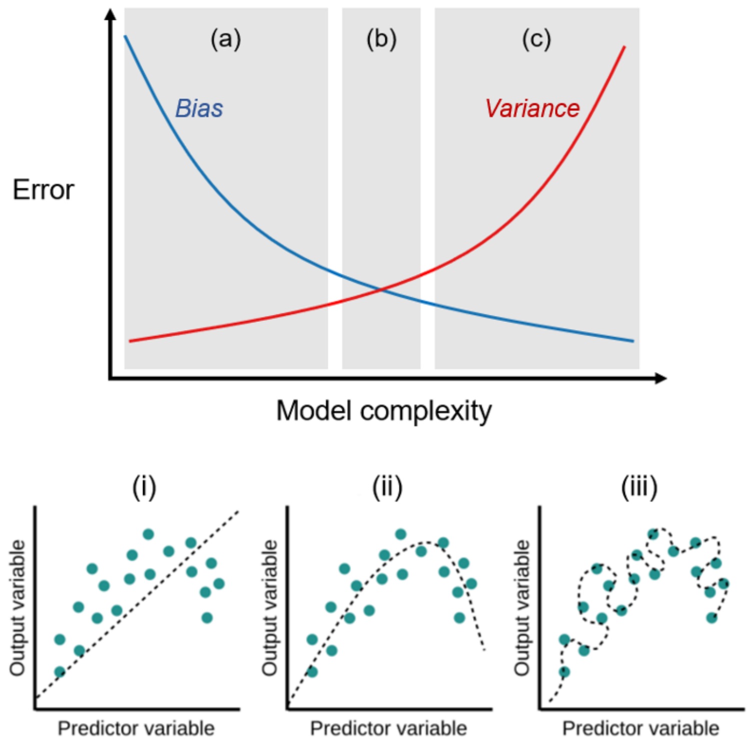

Figure 1: Bias variance trade-off

4. In terms of the bias-variance decomposition, a decision tree with 20 leaves tends to have ______________________ than a decision tree with 3 leaves.

   a. higher variance and lower bias.
   b. higher variance and higher bias.
   c. lower variance and lower bias.
   d. lower variance and higher bias.

Part 1: Bernie Sanders and Donald Trump tweets.

For the first part of the lab, you will work with a dataset containing a sample of tweets from Donald Trump and Bernie Sanders. The objective is to explore how accurately we can predict who the author of a given tweet is based on its content, and to identify which words are the most discriminative.

The tweets have been preprocessed & cleaned for you, and are stored in a so-called document-term matrix format with rows indicating tweets, and columns indicating the frequency of words in different tweets.

1. Begin by importing the file "`trumpbernie_2016.csv`" (hint: for example by using `data.table`'s `fread()` function or the standard `read.csv()`). Report how many *rows* and *columns* there are in this dataset (hint: you may for example use `dim()`). Would you characterize this data set as being *high-dimensional* or *low-dimensional*? Based on this, do you expect that a standard logistic regression will work well for the purpose of prediction?

2. Estimate a *standard logistic regression model* on the whole data set using the `glm` function (recall to specify `family="binomial"` in `glm`). The outcome variable is `trump_tweet`, and the rest of the columns (the word frequencies) are the input variables. Note: estimating the model may take a couple of minutes. When the estimation of the model is finished, do the following:

   a. Extract the coefficients from the estimated model using the `coef()` function and inspect the coefficients that are placed 1010–1050 in the output from `coef()` (hint: to inspect the coefficients placed 1010–1050 you may use standard brackets, e.g., `coef(mymodel)[1010:1050]`). Do you notice anything special?

   b. Examine the *training* accuracy of the estimated model. What does this result suggest about the predictive capacity of the model? You may use the following code to compute the training accuracy:

```r
# Extract predictions on training data & observed values
comparison_df <- data.frame(train_predictions=mymodel$fitted.values,
                            observed=mymodel$y)
# Apply prediction threshold
comparison_df$train_predictions<-ifelse(comparison_df$train_predictions>=0.5,
                                        yes = 1,
                                        no = 0)
# Compute accuracy (scale: 0-1, 0=0%, 1=100%)
nrow(comparison_df[comparison_df$train_predictions==comparison_df$observed,]) /
  nrow(comparison_df)
```

3. Use the `caret` package to implement a 3-fold cross-validation procedure that estimates the *test* accuracy of a *standard logistic regression* (hint 1: two functions are relevant here: `trainControl` and `train` | hint 2: specify `method="glm"` and `family='binomial'` in `train` to fit a standard logistic regression). Note, before you run `train()`, make sure you have formatted the outcome variable `trump_tweet` as a `factor` variable (this is needed for the "caret"'package to recognize the problem as a classification problem). Report the accuracy. Does this result align with your expectations from #1 and #2? Do the results from #2 and #3 provide any indications of either over- or underfitting?

4. Now we shall move beyond the standard logistic regression, and more specifically, turn to ridge regression for our prediction task. This importantly entails deciding on a value for the parameter λ. Use `glmnet`'s function `cv.glmnet` to find the λ that minimizes the test error, and report the associated test accuracy. Is this a better or worse prediction model compared to the one in #2–3? Which of the two models do you believe have a higher variance? Why?

   - When specifying `cv.glmnet`'s arguments, note the following:

i. Unlike `glm` and `train`, the `cv.glmnet` function does not have a `data` argument. Instead, you have to specify `x` and `y` separately (hint: you can for example use `$` to extract and delete columns).
ii. `alpha` dictates the model type (0=ridge, 1=lasso).
iii. Finally, set the number of folds to 5 (`nfolds=5`), `family`=”binomial” (to indicate that `y` should be treated as a binary variable), and `type.measure="class"` (to retrieve a measure of *accuracy*).

- OBS: before using ridge regression, we must standardize X. You can do so using R’s `scale()` function, e.g., `X <- scale(X)`.

5. Plot lambda against the misclassification error (hint: just `plot()` the object which you stored the output from `cv.glmnet`). Interpret the plot in terms of *bias* and *variance*. Note that the x-axis is plotted on a log scale; that explains why you *may* see negative values (negative logged values correspond to very small values on the original scale).

6. Lastly, extract the coefficients associated with the lowest test error (hint: you can use `coef(myfit, s='lambda.min'` for this). Have a closer look at the coefficients with the largest *positive* and largest *negative* values. What do they reveal? Do the words you find on either side confine to your expectations?

Part 2: Social Network Ad Purchase.

For the second part of the lab, you will work with a dataset that contains information about individuals’ purchasing behavior under exposure to online ads. The data originates from an online shopping site, and contains sociodemographic variables `Age`, `Gender` and `Salary`, as well as 20 variables ($X_1, X_2, \ldots, X_{20}$) that reflect digital traces left by the individuals. Our goal is to examine how well we can predict purchases (`Purchase=0` or `Purchase=1`) on the basis of these variables, contrasting three methods: *logistic regression*, *kNN*, and *decision trees*. The people providing us with the data suggest that *some* of digital trace variables could be very relevant for predicting `Purchase`, but they also note that many of them likely are irrelevant. However, they do not know which is which.

1. Begin by importing the file “`Kaggle_Social_Network_Ads_Augmented.csv`”. Format the outcome variable `Purchased` as a `factor` variable (this is required for the subsequent analysis using the `caret` package). You may also delete the `user_id` column, as it will not be used.

2. Use the `caret` package to implement a 10-fold cross-validation that assesses the test accuracy of a *standard logistic regression model*. Report its test accuracy.
   - Hint (i): two functions are relevant here: `trainControl` and `train`.
   - Hint (ii): to use a logistic regression model with *caret*’s `train` function, specify: `model="glm"` and `family="binimial"`.
   - Note: To ensure that the folds generated by `caret` are identical for different models, add a `set.seed(12345)` above each use of `train()`.

3. The second method you will use to predict ad purchase is the *k-nearest neighbors* algorithm. As we talked about in the lecture, it has one key parameter, k, that the researcher must set. Use the `caret` package to implement a 10-fold cross-validation procedure that examines the test accuracy of *kNN* under different choices of k (hint 1: set `method="knn"` to estimate a *kNN* with `caret`| hint 2: use the `tuneGrid` argument to specify your grid of k values). Consider the following values of k: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 50, 100\}$. Because *kNN* is sensitive to the scale of variables, we must standardize our variables prior to estimation (hint: you can add the following argument to `train()`: `preProcess=c("center","scale")`, to do so). From the results, plot the *test accuracy* against k (hint: results can be extracted from the `train` object by using `$results`). Interpret this plot. How does the performance—for what you deem to be the best k—compare to that of the *standard logistic regression*? Speculate why the method you found to have better performance did so.

4. Motivated by the results obtained in #3, we want to explore an alternative method. Based on the properties—which we discussed in the lecture—of *decision trees*, do you think it has the potential to perform better than *kNN*? If yes, why?

5. Estimating decision trees is what you will do now. Decision trees have one key parameter that a researcher must choose prior to estimation: the complexity parameter α (in `caret`: `cp`). Please do the following:

   a. Use the `caret` package (`method="rpart"`) to implement a 10-fold cross-validation procedure to assess how the test accuracy changes as a function of the complexity parameter (ISL: α; here: `cp`). Hint: use the `tuneGrid` argument to specify your grid of *cp* values. Consider the following values for *cp*: $\{0, 0.01, 0.025, 0.05, 0.10, 0.25, 0.5, 1.0\}$. Plot the *test accuracy* against *cp*. Interpret this plot. How does the accuracy for the best *cp* compare to *kNN*? What do you attribute this difference to?

   b. Use the `rpart.plot` package to plot the decision tree you found to be the best in *b* (hint: you can extract the model with the highest accuracy from a `caret` model using `$finalModel`). Provide an interpretation. Does any of the digital trace variables ($X_1, \ldots, X_{20}$) show up in the tree?

   c. Use `caret`'s `varImp()` function to calculate *variable importance* scores. Interpret.

Part 3: A fictive scenario.

For the third and final part of the lab, you will compare how well OLS and random forest perform on a *simulated* data set. Suppose in this fictive scenario that previous research has studied the phenomena in question using OLS and assumed linearity. You, having taken this course, hypothesize that there may be some interesting non-linearity present. This is what you will explore now.

1. Begin by importing the file "`fictive_scenario_dt.csv`". It contains an outcome variable Y and five input variables $X_1 \ldots X_5$. All variables are continuous.

2. To replicate the state of the art in this imagined literature, use the `caret` package to implement a 10-fold cross-validation that assesses the test error of a *standard linear regression model.* Report its test error (either *RMSE* or R^2).

   - Hint (i): two functions are relevant here: `trainControl` and `train`.
   - Hint (ii): to use a linear regression model with *caret*'s `train` function, specify: `model="lm"`.
   - Note: To ensure that the folds generated by `caret` are identical for different models, add a `set.seed(12345)` above each use of `train()`.

3. To explore your hypothesis that there may be important non-linearities underlying this social phenomena, you shall apply *random forests.* Implement a 10-fold cross-validation procedure using the `caret` package (`method="rf"`). For random forests, a key parameter that the researcher must select is *mtry*, which controls the number of variables that are considered at each split (corresponding to parameter m in ISL). Specifically, you shall consider the following values for *mtry*: $\{1, 2, 3, 4, 5\}$ (hint: again, you specify this grid using the argument `tuneGrid`). Another important parameter of random forest models is the *number of trees* to use. Here you shall use 250 (hint: you may enter `ntree=250` as an argument in `train()` to do so). Once the estimation has finished, please do the following:

   a. Identify which mtry value results in the best predictive performance.

   b. Relate the performance of the best random forest model to the standard linear model. Do you find a meaningful improvement using random forest? What does this suggest about the data, and more specifically about the relationship between X and Y?

c. Calculate the *variable importance scores* for both the random forest model and the standard linear regression model. Hint: you can use `caret`'s `varImp()` function to calculate variable importance. It takes as input the caret-model-object. Do you find any differences between the models, or are they in agreement on the most important variables?

d. Create a *partial dependency plot* for the variable that attained the highest importance score for the random forest model. Create this plot both for the linear regression model and the random forest model, and compare. Hint: you can use `pdp`'s `partial()` function to create a partial dependency plot. For it, you need to specify the following arguments: `object` (set to your caret-train-object), `pred.var` (set to the name of the variable you want to plot for), `train` (set to the data you had as input for the caret-train-function), `grid.resolution` (set to 20). Interpret your findings.

Wrap-up (bonus)

For those of you who have reached this point, whether by completing all tasks or by skipping some, we suggest one final task: Reflect on how what you have learned today could be relevant to your own research. To guide this reflection, consider the following sub-questions:

- As social scientists, we often focus on *inference* (emphasizing $\hat{\beta}$). Reflect on how prediction (focusing on $\hat{y}$) might be used in your research. For instance:
  - Could prediction be valuable as an end in itself? A relevant example is the *Fragile Family competition*, which aimed to predict early adult outcomes based on childhood information.[1]
  - Could you use predictions to automate data labeling (as in the text data example) or to extrapolate your model information to new cases (as in the image data example).[2]
- Sticking to the classical social science task of *inference*, consider these questions about your (current and future) datasets:
  - Are all relationships *linear* and *additive*, or could there be complex, less well-understood relationships?
  - What is the *dimensionality* of these data sets?

[1]Relevant reference: https://www.pnas.org/doi/10.1073/pnas.1915006117
[2]Relevant reference: https://journals.sagepub.com/doi/abs/10.1177/00491241221134526