Statistical Learning, Tutorato #1

Veronica Vinciotti, Marco Chierici

March 17, 2023

Exercise 1

In this exercise, you will investigate the use of a k-NN classifier on the Stock Market data used in the lab lesson, and in particular you will investigate the bias/variance trade-off behind the choice of k.

- Split the Stock Market data into a train and test partition, using the same criterion adopted in the lab lesson (temporal split).
- Using `Lag1` and `Lag2` as predictors, fit a k-NN classifier with different values of k (for example, from 1 to 100).
- Evaluate the train and test errors of each model and plot them as a function of $1/k$.

Hints

- To compute the train accuracy, you need to evaluate the knn on the data used for fitting the model. Consider that `class::knn()` does not have distinct functions for fit and predict.
- The plot should be similar to Figure 2.17 in the textbook.
- You may also repeat the exercise using all of the `Lag1` to `Lag5` predictors.

Exercise 2

In this exercise, you will investigate the use of interaction terms within a logistic regression model, in the context of predicting the fate of passengers aboard the RMS Titanic.

- Use the R library "titanic", which contains a `titanic_train` dataframe.
- Select the variables "Pclass", "Sex", "Age" as predictors, and "Survived" as target.
- Explore the data: look especially for missing values (NAs) and remove rows containing them.
- Further split the `titanic_train` into train and test partitions using a simple criterion (e.g., into random halves).
- With a logistic regression model, predict the fate of passengers (Survived) using age as a predictor.
- Plot the probability of survival on a new vector of ages.
- But, "women and children first"! It could be that age is not the only factor affecting survival.
  - Revise the model using both age and gender as predictors; compute model performance on the test set;
  - Revise the model considering an interaction term between age and gender;
  - For both revised models, plot the survival probabilities vs age stratified by gender: compare the plots for the two different models. What happened?

Hints

- To remove rows containing missing values, use the function `na.omit()`.
- Encode the target variable as a `factor`.
- An interaction term between `var1` and `var2` can be added to a model using the notation `var1:var2` in the formula: `target ~ var1:var2`. A convenient way to simultaneously include `var1`, `var2`, and

their interaction as predictors is to use the notation `var1*var2`: `target ~ var1*var2` is equivalent to `target ~ var1 + var2 + var1:var2`.

Exercise 3

In this exercise, you will explore the bias/variance trade-off, that you have studied in Exercise 1 for a classification context, within a regression context. In particular, we use the `Wage` data set from the `ISLR2` library. The dataset includes information about wage and other variables (e.g., age,. marital status, education) for 3,000 male workers in the U.S. mid-atlantic region.

- Load the data set, `attach` it if necessary, and start looking at the variables, as usual.
- Explore the use of *polynomial regression* to predict `wage` using `age`. Consider using a 4th-degree polynomial in the model.
- Use the model you created to predict wages for some values of `age`.
- Plot the data and the predictions, including confidence intervals.
- Explore different degrees for the polynomial: split the dataset into a train and test partition; fit the models on the training set and calculate the (mean-squared) error on the training and test sets. Plot the errors against the degrees, what do you observe?
- Optionally, you can perform the same analysis on the `Auto` data set, included in the `ISLR2` library, to estimate fuel consumption (`mpg` variable) from `horsepower`.

Hints

- `poly(x, N)` creates a polynomial of degree N over the set of points in `x`.
- Build confidence intervals extending +/- 2*SE around the value (SE: standard error).
- The computation of standard error has to be enabled in the `predict()` function.

Exercise 4

The output of regression models and classifiers depends on the choice of predictors that are included in the model. There could well be other predictors (so called *confounders*) that are not in the model and that would capture more clearly the relationship between predictors and response. In this exercise, we explore this aspect using the `Default` dataset from the `ISLR2` library. The dataset contains simulated data with information on 10,000 customers: the goal is to predict whether a customer will default on their credit card debt, given the predictors `student` (No/Yes), `balance` (avg. balance remaining on the credit card), `income` (income of the customer). Recall that `student` is therefore a *dummy variable* (also: *indicator variable*), i.e., a variable taking only binary values (0/1, No/Yes) to represent the absence or presence of something.

- After the usual data exploration, fit a logistic regression model to predict `default` using `student` only. Discuss the results.
- Fit another model to predict `default` using all predictors:
  - Discuss the influence of the predictors on the outcome by examining the model coefficients; compare with the single-predictor results.
  - Use the fitted model to predict the probability of default of a student with a balance=$\$1,500$ and an income of $\$40,000$ (textbook p139 (4.8))
  - Compute the probability of default of a non-student with the same balance and income as above (textbook p139 (4.9).
- To understand the phenomenon of *confounding*, compare the model with student only and the full model. What can you notice about the coefficient for the variable `student` and how can you explain this?
- In order to see it better, reproduce the left and right panel of Figure 4.3.

Hints

- Use the `newdata=list(var1=value1, var2=value2, ...)` argument in `predict()` for passing values to predictors and getting predictions.
- For the left-hand plot of Fig. 4.3, consider the posterior probabilities of your models. Horizontal lines represent the overall default rates (i.e., defaulted to non defaulted ratio) stratified by student status.
- The boxplot function accepts a `formula` argument.