API 222 Problem Set 1

Machine Learning and Big Data Analytics: Spring 2024

Oscar Boochever

Due at 11:59am on February 15 - submit on Gradescope

This problem set is worth 30 points in total. To get full credit, submit your code along with a write-up of your answers. This should either be done in R Markdown or Jupyter Notebook, submitted in one knitted PDF.

Brief survey (0 pts)

Please fill out this brief (ungraded) survey to help Professor Saghafian and the teaching assistants get to know you. The link to the survey can also be found on Canvas under "Quizzes".

Done

Conceptual Questions (15 pts)

1. For each of the following questions, state: (6 pts)

`(1) Whether it is a regression question or a classification question`

`(2) Whether we are interested in inference or prediction`

(a) A health organization is seeking to improve mental health services in a rural area. They want to identify individuals at high risk of developing stress-related disorders. They have demographic data and survey responses about lifestyle and stress levels.

1. Classification. 2. Prediction

(b) In a study exploring gender bias in job recruitment, researchers analyze, using application records and interview feedback, whether female applicants in technology roles are less likely to be called for an interview compared to male applicants.

1. Regression. 2. Inference

(c) A team of researchers is investigating the impact of dietary changes on physical fitness levels among middle-aged adults. They first implement a program promoting a balanced diet and then measure the change in the participants' body mass index (BMI) over six months.

1. Regression. 2. Inference

2. Flexible models versus inflexible models (5 pts)

(a) Flexible models will generally have lower bias than inflexible models. True or False?

True

(b) For the same very large number of observations, inflexible models will likely perform better than flexible models when the number of features is small. True or False?

True

(c) If the underlying data generating process is linear, a flexible model will generally perform worse than an inflexible one. True or False?

True

(d) Non-parametric models impose stronger assumptions on the underlying data generating process than parametric models. True or False?

False

(e) KNN and linear regression are both parametric models, as they both have decision rules. True or False?

False

3. The bias-variance tradeoff (4 pts)

(a) What does bias refer to in the machine learning context?

In a machine learning context, bias can be thought of as the "accuracy" of the model. On a dart board, this would mean centering on the bullseye. In more detail, it is the average difference between predicted and actual y values across all x's.

(b) What does variance refer to in the machine learning context?

In a machine learning context, variance can be thought of as the "consistency" of the model. On a dart board, this would mean the spread of the darts, with low variance being concentrated in one area, and high variance being spread all over. In more detail, it is the average squared difference between each individual prediction and the mean prediction across all possible training datasets. In other words, the amount by which the model would change if we estimated it using a different training set.

(c) Now briefly describe the bias-variance tradeoff.

More flexible models have lower bias, as they more accurately fit the training data. However, this means that they are sensitive to that specific composition of the training data, which means that each cut of training data would produce different models – this means it would have higher variability. This is the fundamental bias-variance tradeoff, which we aim to balance by minimizing MSE (setting first derivative equal to zero).

(d) Briefly explain the issue of overfitting in light of the bias-variance trade-off.

Very flexible models are prone to overfitting, which would result in very low bias, but very high variance.

Data Questions (15 pts)

This dataset focuses on predicting Atherosclerotic Cardiovascular Disease (ASCVD) risk, encompassing clinical, demographic, and lifestyle data. Accurate ASCVD risk prediction is crucial for public health policy, enabling early intervention and informed healthcare strategies. It aids policymakers and health officials in reducing the burden of cardiovascular diseases, a leading global cause of death, and in formulating policies for healthier lifestyles. Utilizing this data in machine learning can lead to improved public health outcomes and more efficient healthcare resource allocation, demonstrating the dataset's significant implications for public health policy and patient care.

For any non-integer numbers, please report your numbers to exactly two decimal places for full credit.

1. Preliminary data exploration (5 pts)

```r
# Load data and libraries
library(tidyverse)

data <- read_csv('heartRisk.csv')
```

(a) How many observations and variables are in the dataset?

```r
dim(data)
```

```
## [1] 1000    9
```

1000 observations and 9 variables

(b) Are any of the columns categorical? If so, which ones?

```r
sapply(data, class)
```

```
##         isMale       isSmoker     isDiabetic isHypertensive            Age
##      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
##       Systolic    Cholesterol            HDL           Risk
##      "numeric"      "numeric"      "numeric"      "numeric"
```

No, they are all numeric.

(c) Compute the mean and standard deviation of the `Risk` score.

```r
data %>%
  summarise('Risk Mean' = round(mean(Risk), 2),
            'SD Risk' = round(sd(Risk), 2))
```

```
## # A tibble: 1 x 2
##   ‘Risk Mean‘ ‘SD Risk‘
##         <dbl>     <dbl>
## 1        19.7      17.0
```

See summary table above.

For the next few questions, set the seed to 222 and randomly put 20% of your observations in a test set and the remaining observations in a training set.

```r
set.seed(222) #to API-222 number so random sample stays the same repeatedly

all_ids <- 1:nrow(data)

test_ids <- sample(all_ids, round(0.2 * nrow(data))) #ids to pull out to become the training data

training_ids <- all_ids[!(all_ids %in% test_ids)]

#test code to see if works equivalently
training_ids_review_section <- which(!(1:nrow(data) %in% test_ids))
identical(training_ids, training_ids_review_section)
```

```
## [1] TRUE
```

```r
# Use ids to create datasets
test_data <- data[test_ids, ]
training_data <- data[training_ids, ]
```

2. When you use your training data to build a linear model that regresses `Risk` on all other features available in the data (plus an intercept), what is your test Mean Squared Error? (2 pt)

```r
# Create "kitchen sink" model and view results
risk_kitchen_sink <- lm(Risk ~ ., data = training_data)
summary(risk_kitchen_sink)
```

```
##
## Call:
## lm(formula = Risk ~ ., data = training_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.7644  -5.2747  -0.6076   3.8325  31.3246
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -85.725276   2.974308 -28.822  < 2e-16 ***
## isMale           4.894463   0.534818   9.152  < 2e-16 ***
## isSmoker         9.165136   0.536084  17.096  < 2e-16 ***
## isDiabetic      10.568583   0.536353  19.705  < 2e-16 ***
## isHypertensive   4.716782   0.535085   8.815  < 2e-16 ***
## Age              0.919675   0.023155  39.719  < 2e-16 ***
## Systolic         0.208531   0.008633  24.156  < 2e-16 ***
## Cholesterol      0.070132   0.013169   5.326 1.31e-07 ***
## HDL             -0.094243   0.011152  -8.451  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 7.539 on 791 degrees of freedom
## Multiple R-squared:  0.7991, Adjusted R-squared:  0.797
## F-statistic: 393.2 on 8 and 791 DF,  p-value: < 2.2e-16
```

```
# Predict values
names(test_data[, 9]) #confirm the variable index for next line
```

```
## [1] "Risk"
```

```
predicted_risk_ks <- predict(risk_kitchen_sink, test_data[, -9])

# Calculate test MSE
test_mse_risk_ks <- round(mean((predicted_risk_ks - test_data$Risk)^2), 2)
test_mse_risk_ks
```

```
## [1] 81
```

```
#Compare test MSE to training MSE (optional)
training_mse_risk_ks <- round(mean((risk_kitchen_sink$residuals)^2), 2)
training_mse_risk_ks / test_mse_risk_ks
```

```
## [1] 0.6938272
```

The test MSE is 81.00.

3. Now use your training data to build a linear model that regresses Risk on only three variables: Age, isDiabetic, and isHypertensive (include an intercept)

(a) What is your test MSE? (1 pt)

```
# Create simpler model and view results
risk_simple <- lm(Risk ~ Age + isDiabetic + isHypertensive, data = training_data)
summary(risk_simple)
```

```
##
## Call:
## lm(formula = Risk ~ Age + isDiabetic + isHypertensive, data = training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.755  -8.348  -1.562   6.092  44.404
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -40.72535    2.19618 -18.544  < 2e-16 ***
## Age              0.87848    0.03556  24.703  < 2e-16 ***
## isDiabetic      10.44891    0.82649  12.643  < 2e-16 ***
## isHypertensive   5.28104    0.82497   6.401 2.63e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 11.64 on 796 degrees of freedom
## Multiple R-squared:  0.5184, Adjusted R-squared:  0.5166
## F-statistic: 285.6 on 3 and 796 DF,  p-value: < 2.2e-16
```

```
# Predict values
names(test_data[, 9]) #confirm the variable index for next line
```

```
## [1] "Risk"
```

```
predicted_risk_simple <- predict(risk_simple, test_data[, -9])

# Calculate test MSE
test_mse_risk_simple <- round(mean((predicted_risk_simple - test_data$Risk)^2), 2)
test_mse_risk_simple
```

```
## [1] 146.78
```

```
#Compare test MSE to training MSE
training_mse_risk_simple <- round(mean((risk_simple$residuals)^2), 2)
training_mse_risk_simple / test_mse_risk_simple
```

```
## [1] 0.9177
```

The test MSE is now 146.78.

(b) Create a table that shows the coefficients from both of your models. Use the `stargazer` package to do this if you are working in an RMD. (2 pt)

```
# Your code here
library(stargazer)
#stargazer(risk_kitchen_sink, risk_simple)
```

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Thu, Feb 08, 2024 - 11:43:09

Table 1:

	Dependent variable:	
	Risk	
	(1)	(2)
isMale	4.894***	
	(0.535)	
isSmoker	9.165***	
	(0.536)	
isDiabetic	10.569***	10.449***
	(0.536)	(0.826)
isHypertensive	4.717***	5.281***
	(0.535)	(0.825)
Age	0.920***	0.878***
	(0.023)	(0.036)
Systolic	0.209***	
	(0.009)	
Cholesterol	0.070***	
	(0.013)	
HDL	-0.094^{***}	
	(0.011)	
Constant	-85.725^{***}	-40.725^{***}
	(2.974)	(2.196)
Observations	800	800
R^2	0.799	0.518
Adjusted R^2	0.797	0.517
Residual Std. Error	7.539 (df = 791)	11.635 (df = 796)
F Statistic	393.184*** (df = 8; 791)	285.600*** (df = 3; 796)
Note:		*p<0.1; **p<0.05; ***p<0.01

(c) Provide some intuition for what it means that some of the coefficients changed between the two regressions. (1 pt)

Some of the variables in the second (short) regression are explainable by or have correlation with variables in the first (longer) regression. Those variables in the long regression but not the short may also have correlation with the dependent variable. In essence, there is omitted variable bias for the independent variables in the short regression, and the coefficients either overstate or understate their true explanatory power compared to when controlling for relevant additional confounders.

4. When you use your training data to build a KNN model that regresses Risk on all other features in the data, what is your test Mean Squared Error with $K = 2$? (1 pt)

```r
library(FNN)
knn_reg1 <- knn.reg(training_data[, -9],
                    test_data[, -9],
                    training_data$Risk,
                    k = 1)

mse_knn1_test <- mean((knn_reg1$pred - test_data$Risk)^2)
mse_knn1_test
```

```
## [1] 221.3577
```

With $K = 1$, the test MSE is 221.36

5. When you use your training data to build a KNN model that regresses Risk on all other features in the data, what is your test Mean Squared Error with $K = 10$? (1 pt)

```r
knn_reg10 <- knn.reg(training_data[, -9],
                     test_data[,-9],
                     training_data$Risk,
                     k = 10)

mse_knn10_test <- mean((knn_reg10$pred - test_data$Risk)^2)
mse_knn10_test
```

```
## [1] 126.5375
```

With $K = 10$, the MSE is 126.54.

6. Between the standard linear regression and the KNN regression, which performed better? (1 pt)

The longer "kitchen sink" linear regression performed best with a lower MSE (81.00) than the best performing KNN(10) model (126.54).

7. From an inference standpoint, which of these models would we rather use? (1 pt)

From an inference standpoint, we would rather use the linear model, as the coefficients are more interpretable. For example, we know the difference in Risk that being male versus female poses, holding all the other variables in our model constant. This has greater interpretability, and can lead to policy relevant actions.