Data Mining

LABDATA

Coordenação:

Prof. Dr. Adolpho Walter Pimazzi Canton

Profa. Dra. Alessandra de Ávila Montini

Disciplina: Machine Learning

Prof. Carlos Eduardo Martins Relvas

Currículo

- Bacharel em Estatística, Universidade de São Paulo.
- Mestre em Estatística, Universidade de São Paulo.
- Doutorando em Ciência da Computação, Universidade de São Paulo.

- Itaú, 2010-2015. Principais atividades:

  - Consultoria estatística para várias áreas do banco com foco principal em melhorias no processo de modelagem de risco de crédito.

  - De 2013 a 2015, participação do projeto Big Data do banco usando tecnologia Hadoop e diversas técnicas de machine learning. Desenvolvemos diversos algoritmos em MapReduce usando R e Hadoop streaming, criando uma plataforma de modelagem estatística no Hadoop.

- Nubank, dede 2015. Equipe de Data Science, responsável por toda a parte de modelagem da empresa, desde modelos de crédito e até mesmo identificar motivos de atendimento.

Agenda

- **Random Forest**
- **Missing**

Random Forest

Bagging

Árvores de decisão são conhecidas por apresentarem alta variância, ou seja, geralmente são instáveis. Sob pequenas mudanças nos dados, as variáveis escolhidas e os pontos de corte podem se alterar completamente.

Para amenizar este problema, podemos utilizar bagging. Consiste na criação de várias amostras de bootstrap (B) e o treino do seu modelo de classificação ou regressão em todas elas.

Bagging

Por que bagging tende a melhorar a performance?

- $E(\bar{X}) = E(X_i), \forall\, i$.

- $Var(\bar{X}) = \rho\sigma^2 + \frac{(1-\rho)\sigma^2}{B}$.

Bagging

Por que bagging tende a melhorar a performance?

- $E(\bar{X}) = E(X_i), \forall\, i$.
- $Var(\bar{X}) = \rho\sigma^2 + \frac{(1-\rho)\sigma^2}{B}$.

0

Random Forest

Evolução do método de bagging de árvores de decisão, criado por Leo Breiman em 2001.

Diminui as correlações das diferentes estimativas de cada árvore. Para isso, durante o processo de criação de cada árvore, apenas algumas variáveis são escolhidas aleatoriamente para cada quebra.

Correlações em Bagging: ~ 0.5
Correlações em Random Forest: ~ 0.05

Random Forest

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

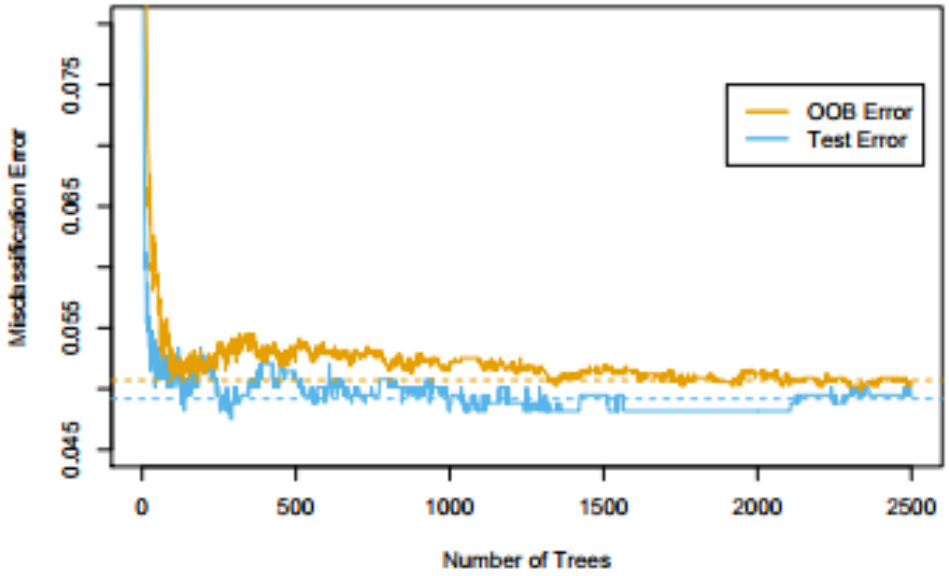
Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

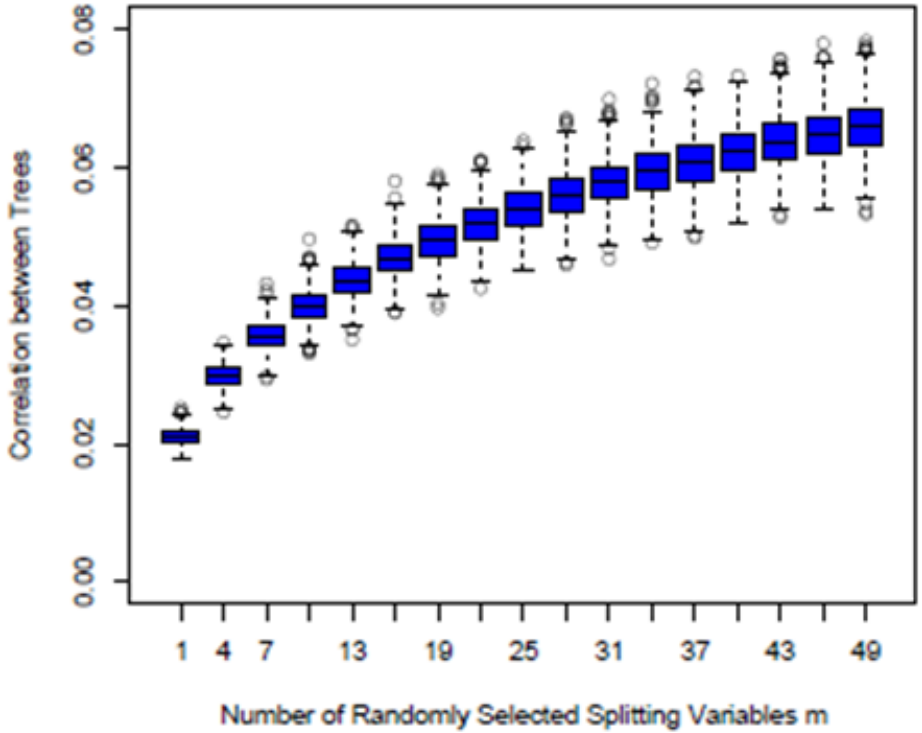
Fonte: The elements of Statistical Learning

Random Forest

Erro Out-of-Bag – Cada árvore gerada não utiliza uma parte dos dados originais (~33%) devido ao procedimento de Bootstrap. Estas observações são avaliadas e o erro é chamado de erro out-of-bag. Este erro é uma estimativa não viesada para o erro na base de teste.

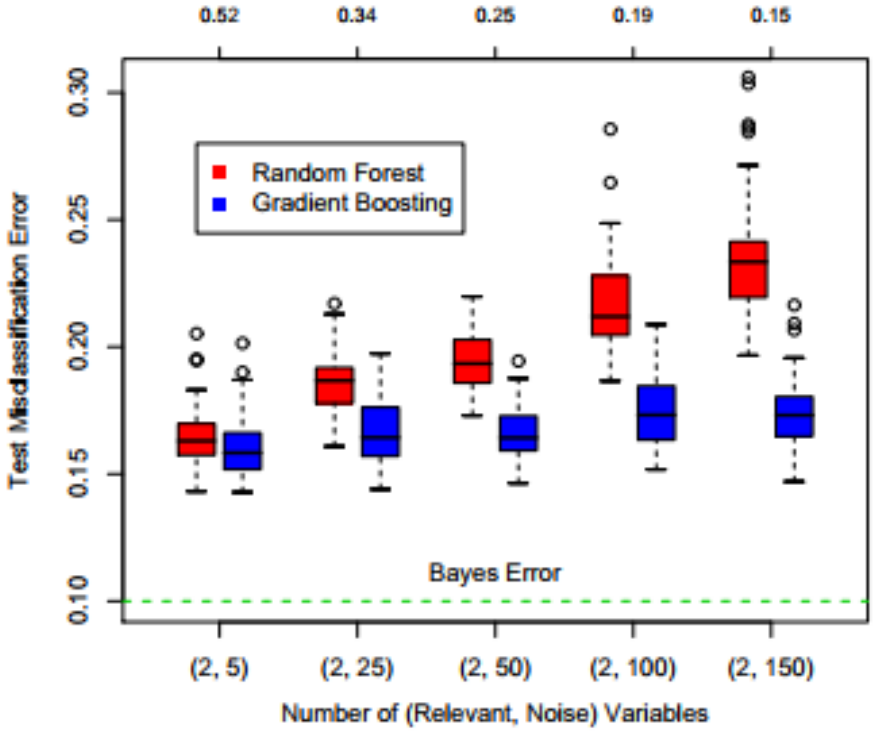


Random Forest



Fonte: The elements of Statistical Learning

Random Forest



Fonte: The elements of Statistical Learning

Random Forest

Por default, temos:

- m é igual a $\lfloor \sqrt{p} \rfloor$ e tamanho mínimo do nó igual a 1 para classificação.
- m é igual a $\left\lfloor \frac{p}{3} \right\rfloor$ e tamanho mínimo do nó igual a 5 para regressão.

Estes parâmetros devem ser otimizados, pois dependem de problema para problema.

Random Forest

Laboratório R - Base de Spam

Base com 4.601 e-mails. Porcentual em que 54 palavras ou pontuações aparecem em cada e-mail. Além disso, temos o tamanho médio das palavras, tamanho da maior palavra e quantidade de palavras.

Disponível em: https://archive.ics.uci.edu/ml/datasets/Spambase

Objetivo:
Criar um detector automático de SPAM que verificará cada novo e-mail.

Random Forest

Laboratório R – Base de Spam

Carregando base e definindo nomes das colunas

Criando bases de desenvolvimento, validação e teste

Treinando um modelo de Random Forest

Avaliando os resultados

Random Forest

Carregando base e definindo nomes das colunas

```
> dados = read.table("spambase.data", sep=",", header=F)
> nomes = c("word_freq_make",  "word_freq_address",  "word_freq_all", "word_freq_3d",
+           "word_freq_our", "word_freq_over", "word_freq_remove", "word_freq_internet",
+           "word_freq_order", "word_freq_mail", "word_freq_receive", "word_freq_will",
+           "word_freq_people", "word_freq_report", "word_freq_addresses", "word_freq_free",
+           "word_freq_business", "word_freq_email", "word_freq_you", "word_freq_credit",
+           "word_freq_your", "word_freq_font", "word_freq_000", "word_freq_money",
+           "word_freq_hp", "word_freq_hpl", "word_freq_george", "word_freq_650",
+           "word_freq_lab", "word_freq_labs", "word_freq_telnet", "word_freq_857",
+           "word_freq_data", "word_freq_415", "word_freq_85", "word_freq_technology",
+           "word_freq_1999", "word_freq_parts", "word_freq_pm", "word_freq_direct",
+           "word_freq_cs", "word_freq_meeting", "word_freq_original", "word_freq_project",
+           "word_freq_re", "word_freq_edu", "word_freq_table", "word_freq_conference",
+           "char_freq_pvir", "char_freq_par", "char_freq_bra", "char_freq_exc",
+           "char_freq_dolar", "char_freq_num", "capital_run_length_average",
+           "capital_run_length_longest", "capital_run_length_total", "SPAM")
> names(dados) = nomes
```

Random Forest

Criando bases de desenvolvimento, validação e teste

```
> set.seed(432)
> id <- sample(1:nrow(dados), nrow(dados)*0.8)
> id.des <- sample(id, nrow(dados)*0.7)
> id.val <- id[!(id %in% id.des)]
> dados.des <- dados[id.des,]
> dados.val <- dados[id.val,]
> dados.test <- dados[-id,]
```

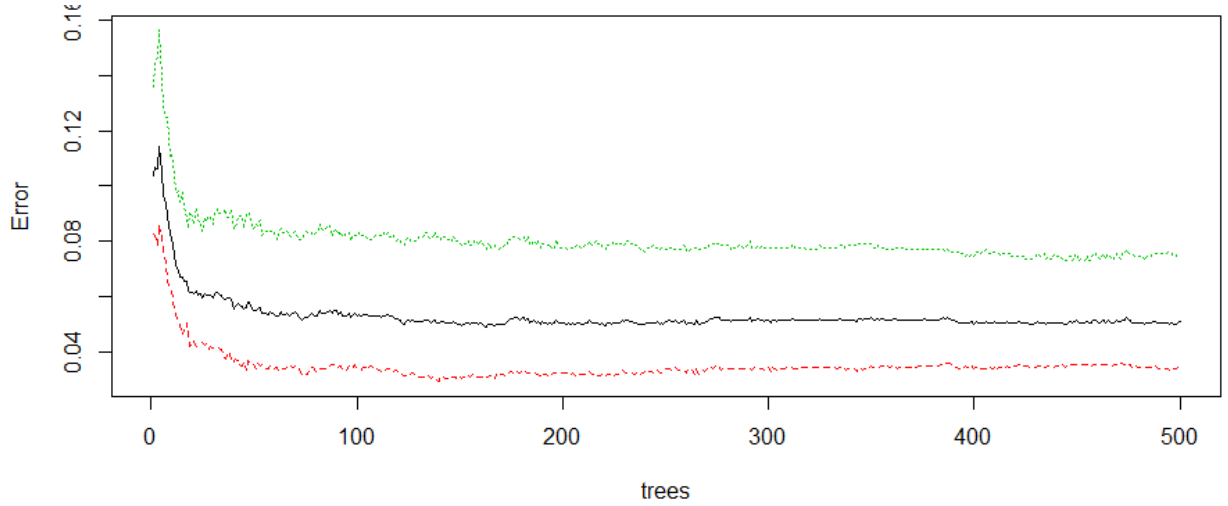
Por que precisamos de uma base de validação neste caso?

- Usamos esta base de validação para otimizarmos alguns parâmetros do Random Forest.

Random Forest

Treinando um modelo de Random Forest

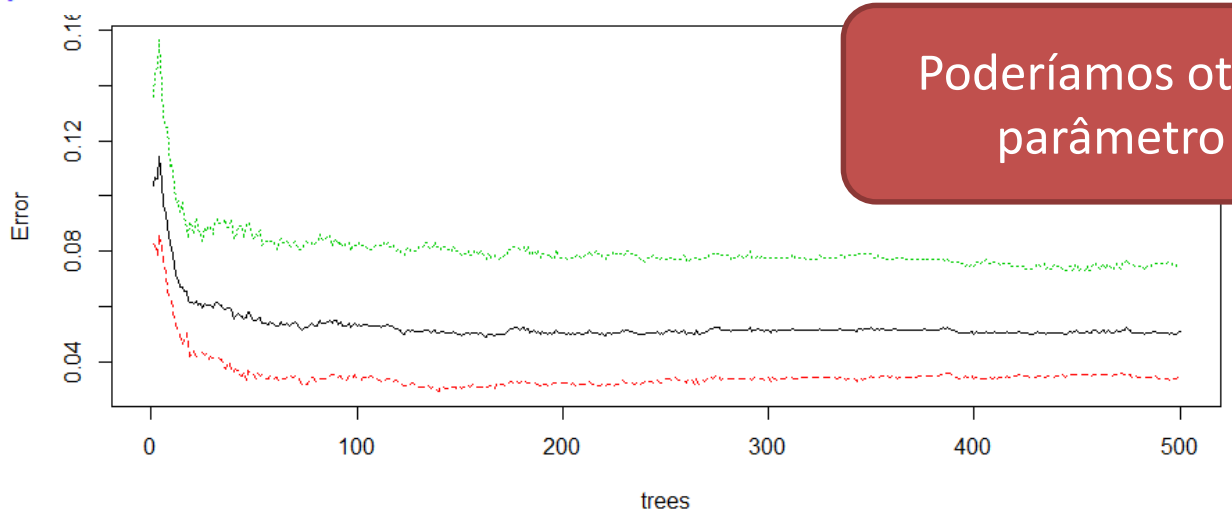
```
> formula <- paste0("as.factor(", nomes[58], ") ~ ", paste0(nomes[1:57], col
lapse="+"))
> rf <- randomForest(as.formula(formula), data=dados.des, ntree=500)
> plot(rf)
```



Random Forest

Treinando um modelo de Random Forest

```
> formula <- paste0("as.factor(", nomes[58], ") ~ ", paste0(nomes[1:57], col
lapse="+"))
> rf <- randomForest(as.formula(formula), data=dados.des, ntree=500)
> plot(rf)
```



Poderíamos otimizar o parâmetro “m”

Random Forest

Avaliando os resultados

```
> fit.val = predict(rf, newdata=dados.val)
> fit.test = predict(rf, newdata=dados.test)
>
> sum(ifelse(fit.val!=dados.val[,"SPAM"],1,0))/nrow(dados.val)
[1] 0.04565217
> sum(ifelse(fit.test!=dados.test[,"SPAM"],1,0))/nrow(dados.test)
[1] 0.06080347
```

Mais de 30% de melhora na performance.

Random Forest

Base de dados (“cancer.data”) com 699 observações e 10 variáveis de pacientes com tumores. O objetivo é detectar com base em algumas informações dos tumores se é benigno ou maligno.

Remova os dados missing por meio do comando “na.omit”. Poderíamos fazer algo melhor?

Variáveis:

- 1. Sample code number id number
- 2. Clump Thickness 1 - 10
- 3. Uniformity of Cell Size 1 - 10
- 4. Uniformity of Cell Shape 1 – 10
- 5. Marginal Adhesion 1 - 10
- 6. Single Epithelial Cell Size 1 - 10
- 7. Bare Nuclei 1 - 10
- 8. Bland Chromatin 1 - 10
- 9. Normal Nucleoli 1 - 10
- 10. Mitoses 1 - 10
- 11. Class: (2 for benign, 4 for malignant)

Random Forest

1.) Crie bases de desenvolvimento (60%), validação (20%) e teste (20%). Utilize seed de 432. Monte a formula para ser usado nos modelos com todas as variáveis explicativas.

2.) Construa um random forest com 100 árvores. Isto é suficiente ou precisamos de mais árvores?

3.) Varie o parâmetro mtry usando 1, 2, 5 e 10. Escolha o melhor na base de validação.

4.) Avalie os resultados em todas as bases. Tem ganho em relação a usar apenas uma única árvore?

Dados missing

Dados missing

- Dados missing ocorrem quando não temos informação (variáveis explicativas) preenchida para algumas observações.

- Na prática, quase nunca trabalhamos com bases de dados sem a presença de dados missing.

- É um problema em aberto com vários artigos publicados ano após ano de qual maneira proceder.

Dados missing

- Temos três principais mecanismos de dados missing:

1. **Missing completamente aleatório (MCAR):** ocorre quando os dados missing são completamente aleatórios, o que não não viesa as análises, sendo o tipo mais de mais fácil solução.

   Ex.: Imagine que por uma falha do sistema, perdemos a informação de renda para 10% das observações de forma aleatória.

Dados missing

- Temos três principais mecanismos de dados missing:

2. **Missing aleatório (MAR):** temos este tipo de missing quando o missing não é aleatório, mas está relacionado a variáveis que temos acesso.

   Ex.: Temos observações missing de pessoas preenchendo quantos banheiros tem em suas casas, mas sabemos que o fato das pessoas preencherem ou não está relacionado com a renda. E temos a informação de renda de cada pessoa.

Dados missing

- Temos três principais mecanismos de dados missing:

3. **Missing não aleatório (MNAR):** ocorre quando os dados missing ocorrem pela própria razão do dado ser missing.

   Ex.: É conhecido que dependendo da sua renda, a proporção de preenchimento da informação de renda varia. Pessoas mais ricas preenchem menos ou mais pobres preenchem menos.

Dados missing

- Temos três principais mecanismos de dados missing:
- Na prática é muito díficil saber com qual situação estamos lidando.
- Para o mecanismo MCAR, podemos usar alguma técnica básica de imputação (média) que não teremos vieses.
- Para o MAR, podemos construir um modelo predizendo quais são os valores dos dados faltantes.
- Por fim, para o MNAR, não existe solução.

Dados missing

- Formas de tratamento de missing:

1. Remover variáveis com **alta** proporção de missing.

   Só é indicada quando as informações preenchidas não ajudam a prever a variável resposta. Uma alternativa é criar uma dummy (missing X não missing).

2. Remover observações com missing.

   Isso quase nunca é indicado, a não ser em situações nas quais se tem certeza que a observação se deve a um erro.

Dados missing

- Formas de tratamento de missing:

3. Categorizar (step function):

   Uma alternativa é categorizar as variáveis com valores faltante e consider os dados missing como uma categoria.

4. Imputação por uma medida central:

   Podemos imputar a média ou a mediana da distribuição para os valores faltantes. É a estratégia mais simples e mais utilizada no mundo de machine learning. Mas qual o problema com esta estratégia?

Dados missing

- Formas de tratamento de missing:

5. Imputação por um valor aberrante.

   Podemos imputar os valores faltantes por um valor bem diferentes, como, por exemplo, -9999. Esta estratégia só funciona para modelo que envolvem árvores.

6. Imputação por modelo.

   Outra opção é construir modelos preditivos (target seria a variável com missing e variáveis explicativas seriam todas as outras) e substiuir cada missing pelo seu valor previsto. O que acontece se temos 2.000 variáveis com missing?

Dados missing

- Formas de tratamento de missing:

7. Algoritmo EM:

   O algoritmo EM permite para modelos que podemos supor a distribuição normal (regressão linear ou logística), o ajuste do modelo usando dados missing.

8. Multiple Imputation:

   Umas das técnicas mais recomendadas, embora não muito utilizada pelo alto custo computacional. Idéia simular ao bagging. Para cada variável com valor ausente, sorteamos da distribuição observada M vezes, gerando M bases diferentes. Os M modelos são então treinados e utilizamos a média de todos.

Laboratório

Exemplos:

Titanic

Prever a probabilidade de sobrevivência dos passageiros do Titanic utilizando métodos de imputação.

Knowledge • 3,464 teams

Titanic: Machine Learning from Disaster

Fri 28 Sep 2012 — Sat 31 De

Dashboard

Home

Data

Make a submission

Competition Details » Get the Data » Make a submission

Predict survival on the Titanic using

Titanic

```
VARIABLE DESCRIPTIONS:
survival        Survival
                (0 = No; 1 = Yes)
pclass          Passenger Class
                (1 = 1st; 2 = 2nd; 3 = 3rd)
name            Name
sex             Sex
age             Age
sibsp           Number of Siblings/Spouses Aboard
parch           Number of Parents/Children Aboard
ticket          Ticket Number
fare            Passenger Fare
cabin           Cabin
embarked        Port of Embarkation
                (C = Cherbourg; Q = Queenstown; S = Southampton)
```

Dados Missing

Exercícios: http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29

UCI

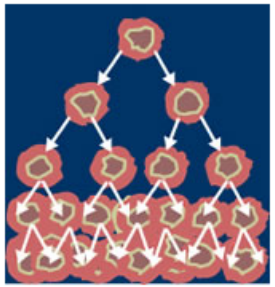
Machine Learning Repository

Center for Machine Learning and Intelligent Systems

Breast Cancer Wisconsin (Diagnostic) Data Set

Download: Data Folder, Data Set Description

Abstract: Diagnostic Wisconsin Breast Cancer Database



Data Set Characteristics:	Multivariate	**Number of Instances:**	569	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	32	**Date Donated**	1995-11-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	390512

Dados Missing

1.) Utilize seed de 42 e crie amostras de treino (70%) e teste (30%).

2.) Ajuste vários modelos alterando a forma de imputação dos dados.

3.) Interprete os resultados.

Referências Bibliográficas

- Hastie, T., Tibshirani, R. & Friedman, J.H. (2001) “The Elements of Statistical Learning”
- Bishop, C.M. (2007) “Pattern Recognition and Machine Learning”
- Mitchell, T.M. (1997) “Machine Learning”
- Abu-Mostafa, Y., Magdon-Ismail, M., Lin, H.T (2012) “Learning from data”
- Theodoridis, S., Koutroumbas, K., (2008) “Pattern Recognition”
- Kuhn, M., Johnson, K., (2013) “Applied Predictive Modeling”
- Burns, P. (2011) “The R inferno”