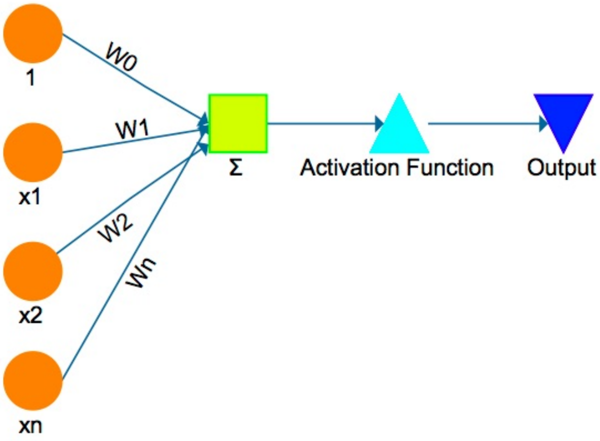
Handbook of Data Science Interview Questions

Statistical Machine Learning

Deep Learning

ML pipelines in Python

Bayesian Machine Learning



SHLOMO KASHANI
M.SC, QMUL, LONDON

Typeset by the author using LaTeX, Tel-Aviv, Israel.

Available in hardcover (ISBN:) and eBook (ISBN:)

WWW.DEEP-ML.COM

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages

First printing, June 2017

CONTENTS

Contents . . . iii

Preface . . . iv

Editor's Note . . . vii

I. Logistic Regression . . . 1

PREFACE

This is the introduction and contents list for the forthcoming book, ***Handbook of Data Science, Machine Learning and Deep Learning Interview Questions*** [1].

In this practical data science interview preparation handbook [2], you will face challenging interview questions in Data Science, Statistical Machine Learning, Probability, Statistics and Deep Learning. ***I have dedicated countless hours after work assembling this booklet, hoping to help candidates excel at their data science interview***.

This booklet contains numerous data science interview questions. The focus lies primarily on developing an understanding of the principles and concepts underlying practical data science. I have not chased mathematical harshness and pureness, but in its place emphasize practical everyday use (for a more scientifically pure analysis, please refer to the bibliography). Although machine learning is applied in a very broad range of domains, this booklet focuses on the application in RTB, Ad-tech, User Retention, CyberSecurity, Quantitative Finance, Kaggle-competitions and Computer Vision.

While theoretical studies are important, practical experience is also invaluable. ”Data Science”,

[1] Thanks to Edward R. Tufte for his inspiration.

[2] With applications in RTB, Ad-tech, User Retention, CyberSecurity, Quantitative Finance, Kaggle-competitions and Computer Vision.

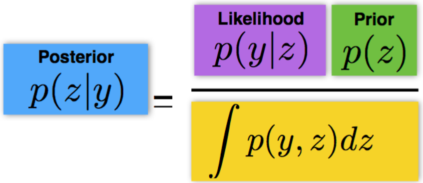


Figure 1: Bayes Rule

a new paradigm in mathematical sciences combining machine-learning, Beyesian Inference [3], statistics and computing with large data sets, is now a basis of employment for numerous mathematics majors. RTB, Finance, Computer Vision and economics are leaders in utilizing data science due to the noticeable value in exploiting large data sets. Data Science has become an extremely active and exciting area with an ever expanding inventory of practical results.

Many advances were enabled by recent developments in the underlying theory and the availability of modern ML pipelines. A notable example is Monte Carlo simulation and MCMC [4] which is one of the most widespread mathematical approaches for computing posterior distributions in Bayesian Machine Learning. There are now four different production grade Python packages targeted at computing the posterior distribution either using MCMC or Variational Inference.

It is equaly directed at both industry practitioners and students and is largely based on hundreds of interviews that the authors conducted either as an interviewer or interviewee. After attempting all questions in this booklet, you will have a much better idea of your week spots. You should attempt all problems and clearly indicate answers, since during a real interview, solutions given with little or no justification may receive little or no credit from the prospective employer. if you are eager to significantly enhance your understanding of Data Science, practicing interview questions and participating in **Kaggle Competitions** and **writing Python/R code** is the way to go. Make sure you are **well prepared by practicing as many coding problems as possible**.

The best way to prepare for an interview is to participate in as many interviews as possible. Just by doing that, you will substantially improve your interview coping skills.

SHLOMO KASHANI.

[3] See: "Deep Probabilistic Programming with Edward. Dustin Tran, Matt Hoffman, Kevin Murphy, Eugene Brevdo, Rif Saurous, David Blei Columbia University, Adobe Research, Google"

[4] PyMC3, PyBayes, Edward and emcee. See: Salvatier, J., Wiecki, T. V., and Fonnesbeck, C. (2016), Probabilistic programming in Python using PyMC3, PeerJ Computer Science, 2, e55.

About me

I have wide-ranging interests in science, mathematical finance, RTB and machine learning, regarding problems at the intersection of Bayesian machine learning and probabilistic modeling. Working as a data scientist provides me with a sense of pleasure, pride and an enduring personal achievement. As a data scientist, my primary research focus has involved the study of noisy big data and the development of associated statistical methodology. In particular, the substantive focus of my work centres on the design and realization of prediction models. In service of this agenda, I utilize extremely large data sets (Hundreds of Tera-Bytes), use PySpark + Jupyter extensively, accelerate R with C++11/Rcpp, employ Bayesian statistics, and conduct large scale Monte Carlo Simulations.

I began my academic career with a focus on industrial engineering and management, a field of study that combines computer science, economics and statistics. This interdisciplinary undergraduate curriculum provided me with a sturdy foundation in probability, calculus, simulation techniques, random processes and software engineering. With my enhanced interest in statistical signal processing, I decided to apply for the graduate DSP program at the University of London. Yet, the study of DSP has traditionally been the near-exclusive province of students with an electrical engineering background. Lacking such credentials, I was somewhat intimidated; still, I was undeterred and completed the programme with distinction. **Moral: do not give up on becoming a data-scientist**.

EDITOR'S NOTE

[The note below was written by XXX, Chief Editor, *Handbook of Data Science Interview Questions*] vii

"Not everything that can be counted counts, and not everything that counts can be counted."

- Albert Einstein (1879-1955)

CHAPTER I

LOGISTIC REGRESSION

The Sigmoid

Please send your comments/feedback/errata to: shlomo@deep-ml.com

1. Describe a project in which you used Binary Classification.
   (a) What was the purpose of the project?
   (b) What was the ML pipeline that you used?
   (c) What was the size of the data set?
   (d) How many covariates were there?
   (e) Were there any categorical variables involved? if yes, did you have to transform them so that they can be used for classification?
   (f) How did you determine the optimal number of covariates?
   (g) Which classification algorithm(s) did you employ?
   (h) How did you avoid over-fitting?
   (i) Which accuracy measure did you utilize?
   (j) How did you validate your results?

2. The sigmoid (refer to Figure I.1) also known as the logistic function, is widely used in binary classification and as a neuron activation function in artificial neural networks.

$$\sigma(x) = \frac{1}{1+e^{-x}} \in (0,1). \tag{I.1}$$

   With respect to the Sigmoid function:

   (a) Compute its derivative and (optionally) find the relationship between the Sigmoid function and its derivative.

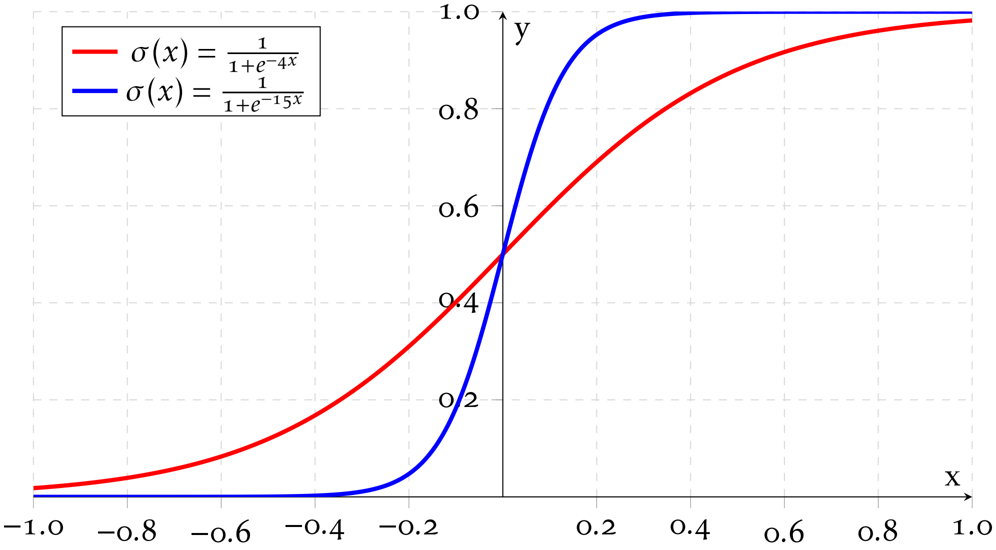


Figure I.1: Sigmoid Function

(b) Show that given $\alpha \to \infty$, $\sigma(\alpha) \to 1$.

(c) Show and given $\alpha \to -\infty$, $\sigma(\alpha) \to 0$.

Answer:

$$\frac{d}{dx}s(x) = \frac{d}{dx}((1+e^{-x})^{-1})$$
$$\frac{d}{dx}s(x) = -1((1+e^{-x})^{(-2)})(e^{-x})(-1)$$
$$\frac{d}{dx}s(x) = ((1+e^{-x})^{(-2)})(e^{-x})$$
$$\frac{d}{dx}s(x) = \frac{1}{(1+e^{-x})^2}(e^{-x})$$
$$\frac{d}{dx}s(x) = \frac{(e^{-x})}{(1+e^{-x})^2}$$

(a) What is the purpose of the following Python code?

Input program 1: *Basic ML*

```
skf = StratifiedKFold(y, n_folds=5, random_state=989,
 ↪  shuffle=True)
```

Answer:

"Cross Validation" is a cornerstone in machine learning, allowing data scientists to take full gain of restricted training data. In classification, effective cross validation is essential to making the learning task efficient and more accurate. A frequently used form of the technique is identified as K-fold cross validation. Using this approach, the full data set is divided into K randomly selected folds, occasionally stratified, **meaning that each fold has roughly the same class distribution as the overall data set**. Subsequently, for each fold, all the other $(K-1)$ folds are used for training, while the present fold is used for testing. This process guarantees that sets used for testing, are not used by a classifier that also saw it during training.

3. Given the logit [1] transformation which forms a linear decision boundary:

$$\log\left(\frac{\Pr(Y=1|X)}{\Pr(Y=0|X)}\right)=\theta_0+\theta^\top X$$

Write the mathematical expression for the decision boundary, e.g. **the hyperplane**.

Answer:

The hyperplane is defined by:

$$\theta_0+\theta_1 x_1+\theta_2 x_2+...=ZERO \tag{I.2}$$

1. With respect to the following code segment:

Input program 2: *ML in Python.*

```python
1  def get_best_xxx(train_X, train_y, modelType):
2  params_lr = {'penalty': ['l2'], 'C': [1,2,5,10,50,500,5000],
3          'solver': ['newton-cg'],
4          'fit_intercept': [False, True]}
5  model_lg = LogisticRegression()
6
7  if modelType=='lr':
8  method=model_lg
9  params=params_lr
10 print 'running grid:' + str(params)
11
12 gscv = GridSearchCV(method, params, scoring='roc_auc', cv=4)
13 gscv.fit(train_X, train_y)
14 for params, mean_score, all_scores in gscv.grid_scores_:
15 print('{:.6f} (+/- {:.6f}) for {}'.format(mean_score, all_scores.std()
    ↪  / 2, params))
16 print('params:{params}'.format(params=gscv.best_params_))
17 print('score:{params}'.format(params=gscv.best_score_))
18 return gscv.best_params_
```

(a) What is the purpose of line 2?

(b) What is the purpose of line 12?

(c) What is the purpose of $cv=4$ in line 12?

(d) In **penalty**, why isn't L1 norm used too?

[1] i.e.: log odds

1. The following C++ code is part of a (very basic) logistic regression implementation module.

Input program 3: *Sigmoid*

```
1  std::vector<double> theta={0.8,0.4,1.2};
2  double sigmoid(double x) {
3          double tmp =1.0 / (1.0 + exp(-x));
4          std::cout << "sig=" << tmp<<std::endl;
5          return tmp;
6  }
7  double hypo(std::vector<double> x) {
8          double z;
9          z=std::inner_product(std::begin(x), std::end(x),
    ↪  std::begin(theta), 0.0);
10         std::cout << "z=" << z<<std::endl;
11         return sigmoid(z);
12
13 }
14 int classify(std::vector<double> x)
15 {
16         return hypo(x) > 0.5f;
17 }
```

2. Explain exactly the purpose of θ.

3. Explain exactly the purpose of line 9 e.g. **inner_product**.

4. Explain exactly the purpose of line 17, e.g. **hypo**.

Odds, Log Odds

1. Logistic regression uses the logit function, which is the logarithm of the "odds.". Describe what is meant by **odds**.

Answer:

Odds is the ratio of success to failure rate of an event.

$$\text{logit}(p) \equiv \log\left(\frac{p}{1-p}\right). \tag{I.3}$$

2. Prove that the logit function and the logistic function are inverses of each other.

Answer:

3. Remember that in logistic regression, the hypothesis function is defined as:

$$
\begin{aligned}
h_\theta(x) &= g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} && \text{(I.4)}\\
&= P(y=1|x;\theta) && \text{(I.5)}
\end{aligned}
$$

I

(a) Assume the coefficients of a logistic regression model with two covariates is as follows: $\theta_0 = -3$, $\theta_1 = 6$, $\theta_2 = -1$ And we have an observation with the following values for the independent variables: $x_1 = 2$, $x_2 = 10$
What is the value of the Logit for this observation?

Answer:

The logit(i.e.Log(odds)) equation is:
$log(Odds) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 = logit = -3 + 6 * 2 + -1 * 10 = -1$

(b) What is the value of the Odds for this observation?

Answer:

$Odds = e^{(\theta_0+\theta_1 x_1+\theta_2 x_2)} = 0.3678794.$

(c) What is the value of P(y = 1) for this observation?

Answer:

$$P(y=1) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2)}}$$

$$= \frac{1}{1+e^{-logit}} = 1/(1+exp(-(-1))) = 0.2689414.$$

The Logit Function and Entropy

1. The entropy H of a single binary outcome with probability p is defined as

$$H(p) \equiv -p\log p - (1-p)\log(1-p). \tag{I.6}$$

(a) Where does H has its max value?

Answer:

This entropy has a max value of log(2) for the probability $p = \frac{1}{2}$ providing maximum uncertainty. A lower entropy is a more predictable outcome, with zero providing full certainty.

(b) What is the relationship between entropy H and the the *logit* function (**Hint**:derive the derivative of$H(p)$)?

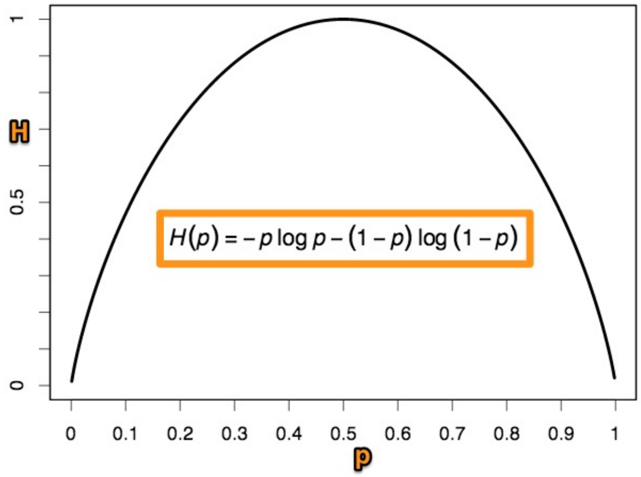


Figure I.2: Binary Entropy

Answer:

The derivative of the entropy with respect to p yields the negative of the logit function:

$$\frac{dH(p)}{dp} = -\text{logit}(p). \tag{I.7}$$

1. The following Python code is part of an automatic differentiation module.

Input program 4: *Entropy*

```
1  import numpy as np
2  import autograd.numpy as np
3  from autograd import grad
4
5  def binaryEntropy (p):
6          return -p*np.log2(p) -(1-p)*np.log2(1-p)
7
8  grad_ent = grad(binaryEntropy)
9  print "binaryEntropy(p) is", binaryEntropy(0.5)
10 print "Gradient of binaryEntropy(p) is", grad_ent(0.5)
```

   (a) Describe the exact value that will be printed on line 9.

   (b) Describe the exact value that will be printed on line 10.

Binary Classification.

1. What is the purpose of the following Python code?

</> Input program 5: *Ml in Python.* </>

```
trainX, testX, trainY, testY = train_test_split(X_df_train_SINGLE,
↪  answers_1_SINGLE, test_size=.33)
```

2. Figure I.3 is an output from running LR for a binary classification task. Describe in your own words the results of the classification.

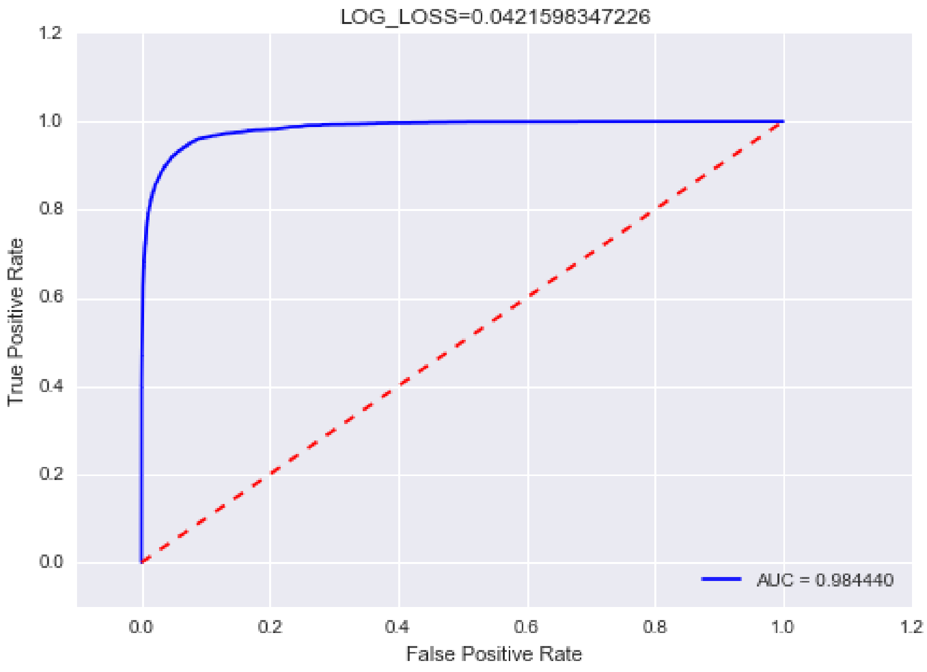


Figure I.3: RUC AUC

1. With respect to the following Python code:

</> Input program 6: *ML in Python.* </>

```
from sklearn.preprocessing import LabelEncoder
from sklearn.pipeline import Pipeline
from collections import defaultdict

d = defaultdict(LabelEncoder)
X_df_train_SINGLE=X_df_train.copy(deep=True)
X_df_train_SINGLE = X_df_train_SINGLE.
apply(lambda x: d[x.name].fit_transform(x))
```

(a) What is its purpose?

(b) Explain what is a LabelEncoder ?

(c) Under which circumstances would you need such code before applying logistic regression?

1. Consider an online advertising application in a Real Time Bidding (RTB) system. Given a user visiting a publisher page, the problem is to select the best advertisement (Ad) for that user. A key element in this matching problem is the click-through rate (CTR) estimation: what is the probability that a given ad will be clicked given some covariates. A Data Scientist collects information related to this RTB system. The following covariates are collected X_1 = time, in MS that a user spent on the web site, X_2 = the size of the Ad in CM.

   Finally Y = is a Binary response variable indicating if a user clicked the Ad. Assume that each response' variable Y_i is a Bernoulli random variable with parameter p_i.. Also assume that

$$p_i = \frac{e^{\alpha+\beta x_i}}{1 + e^{\alpha+\beta x_i}}.$$

The data scientist fits a logistic regression model and produced these estimated coefficients:

$$\hat{\beta}_0 = -6$$
$$\hat{\beta}_1 = 0.05$$
$$\hat{\beta}_2 = 1$$

(a) Estimate the probability that a user who visited the web page for 40 Milli-seconds and is presented with an Ad sized 3.5 CM, clicks on the Ad.

Answer:

$$\hat{p}(X) = \frac{e^{-6+0.05X_1+X_2}}{(1 + e^{-6+0.05X_1+X_2})} = 0.3775$$

(b) How many Milli-seconds would the user in part (a) need to spend to have exactly a 50% chance of clicking the above-mentioned AD?

Answer:

$$\frac{e^{-6+0.05X_1+3.5}}{(1 + e^{-6+0.05X_1+3.5})} = 0.5$$

Which is equivalent to:

$$e^{-6+0.05X_1+3.5} = 1.$$

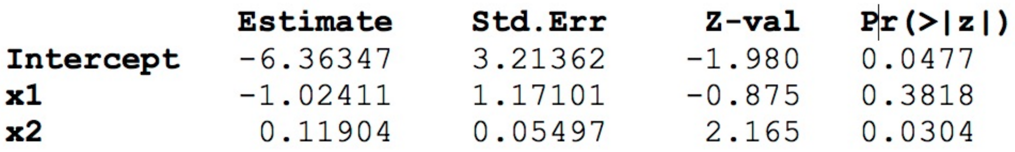
By applying the logarithm function to both sides, we get: $X_1 = \frac{2.5}{0.05} = 50$.

Truly Understanding Logistic Regression.

1. To study factors that affect the CTR of a user on advertisement, a data-scientist collected data from 200K users. The data-scientist fit a logistic regression model with an indicator of a second click within a time frame of one day (1 = click; 0 = no click) as the binary response variable. There are two co-variates:

   x1 = 1 if the user is from the US ; 0 otherwise.
   x2 = income (0 to 100)

   The output from running LR is as follows:



```
             Estimate     Std.Err      Z-val    Pr(>|z|)
Intercept    -6.36347     3.21362     -1.980    0.0477
x1           -1.02411     1.17101     -0.875    0.3818
x2            0.11904     0.05497      2.165    0.0304
```

Figure I.4: Running LR using R .

(a) Using x1 and x2, what are the **odds** of a user clicking an Ad for a second time?

Answer:

$$exp(-6.36 - 1.02 * x1 + 0.12 * x2)$$

(b) A new observation arrives for a user that is from the US and has an income of 100. Caculate the **probability** of a second click for that user.

Answer:

$$z = -6.36 - 1.02 * 1 + 0.12 * 100 = 4.62$$

$$P(1) = exp(z)/(1 + exp(z)) = 0.99.$$

(c) For users that are from the US, is high income associated with an increased probability of a second click? State why.

Answer:

Yes. The coefficient for income is positive (0.11904) and the p-value is less than 0.05 (0.0304).

(d) Is there statistical evidence that living in the US is directly associated with a reduction in the probability of a second click? State why.

Answer:

No. The p-value for this predictor is $0.3818 > 0.05$.