# CSYE 7245–
# Big-Data Systems and Intelligence Analytics
# Practice Exam Solutions

Student Name: ________________________________________

Professor: Nik Bear Brown

Rules:

1. NO COMPUTER, NO PHONE, NO DISCUSSION or SHARING.
2. Ask if you don't understand a question.
3. You may five 8½"×11" sheets of notes (you may use both sides, written or printed as small as you like).
4. Time allowed. 1 hour 30 minutes.
5. Bring pen/pencil. The midterm will be written on paper.

Q1 (5 Points) Two cards are drawn successively and without replacement from an ordinary deck of cards (52 well-shuffled cards). Find the probability that: (a) the first card is an ace but the second is not. (b) at least one card is a diamond. (c) the cards are not the same suit. (d) not more than one card is a face card.

Solution:

$$\text{(a)}\ \left(\frac{48}{51}\right)\left(\frac{4}{52}\right)=\frac{16}{221}\quad \text{(b)}\ 1-\left(\frac{38}{51}\right)\left(\frac{39}{52}\right)=\frac{15}{34}\quad \text{(c)}\ 1-(4)\left(\frac{12}{51}\right)\left(\frac{13}{52}\right)=\frac{13}{17}\quad \text{(d)}\ 1-\left(\frac{11}{51}\right)\left(\frac{12}{52}\right)=\frac{210}{217}$$

Q2 (5 Points) What is the probability of getting exactly two heads after flipping two coins?

Solution:

If one is head and 0 tails:
0 0
0 1
1 0
1 1

There is one event of getting exactly 2 heads
{1 1}

So 1/4 is the probability of getting exactly 2 heads after flipping two coins.

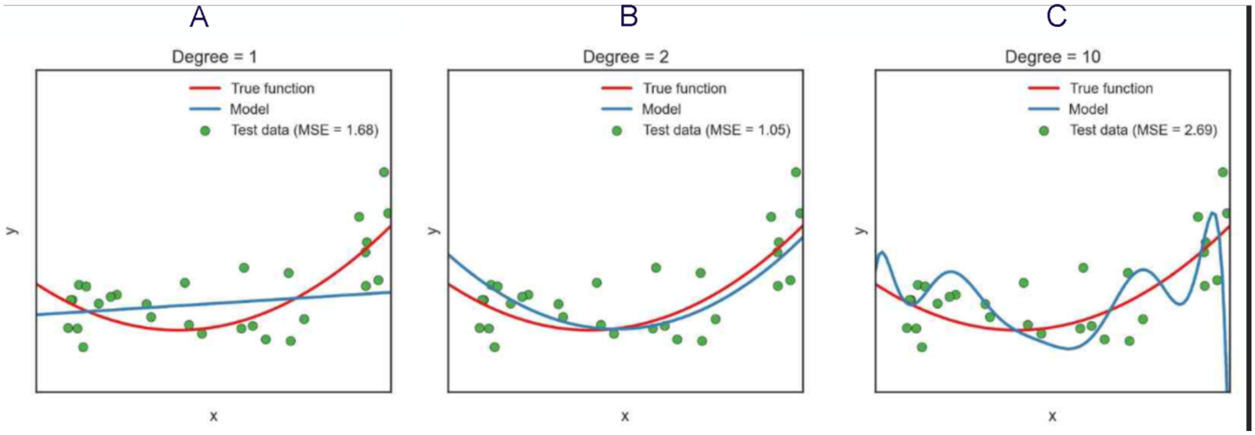
We can also the formula for a discrete random variable based on a binomial distribution:

$$\binom{n}{k} p^k (1-p)^{n-k}$$

X = 2, with probability $\binom{2}{2}(\frac{1}{2})^0(\frac{1}{2})^2 = (\frac{1}{2})^2 = \frac{1}{4}$

$$\frac{4}{52} + \frac{13}{52} - \frac{1}{52} = \frac{16}{52} = \frac{4}{13}$$

Q3 (5 Points) The models A, B and C were fit to the same data. Are any of the three models below underfitting or overfitting?



Solution:

Model A is clearly underfitting as one sees the model is too simple to fit the test data as well as model B.
Model C is clearly overfitting as one sees the poor fit on test data as compared to model B.
Model B is unknown as one would need to test a slightly more complex, say degree 3, model to know if it is underfitting.

Q4 (5 Points) Arrange the following functions in increasing order of asymptotic growth:

- log n
- √n
- $5n^5$
- n log n
- $n^{0.1}$

Solution:

- log n, $n^{0.1}$, √n, n log n, $5n^5$

See http://en.wikipedia.org/wiki/Big_O_notation#Orders_of_common_functions

Q5 (5 Points) Master Theorem: For the following recurrence, give an expression for the runtime T(n) if the recurrence can be solved with the Master Theorem. Otherwise, indicate that the Master Theorem does not apply.

a. $T(n) = 2T(n/2) + n^2$
b. $T(n) = T(n/2) + n$
c. T(n) = T (n/3)− n3

Solution:

a. $T(n) = 2T(n/2) + n^2$

(Case 3) k= log(2)/log(2)=1 f(n) is $n^2$ which is $\theta(n^2)$ so 1<2 and this is Case 3.
Therefore Case 3 is θ(f(n)) which is $\theta(n^2)$

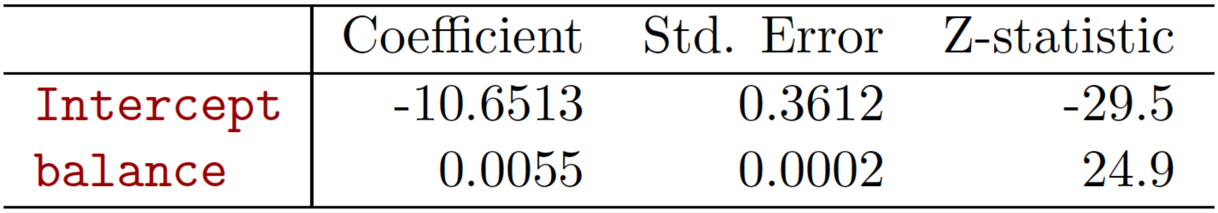
b. $T(n) = T(n/2) + n$

(Case 3) k= log(1)/log(2)=0 f(n) is n which is θ(n) so 0<1 and this is Case 3.
Therefore Case 3 is θ(f(n)) which is θ(n)

c. T(n) = T (n/3)− n3

Does not apply (f(n) is not positive)

Q6 (5 Points) Assume logistic regression is being used to predict whether someone will default on a loan and an independent variable called balance is the only independent variable in the model. It returns the stats below. Is balance a significant predictor of default?



| | Coefficient | Std. Error | Z-statistic |
|---|---|---|---|
| Intercept | -10.6513 | 0.3612 | -29.5 |
| balance | 0.0055 | 0.0002 | 24.9 |

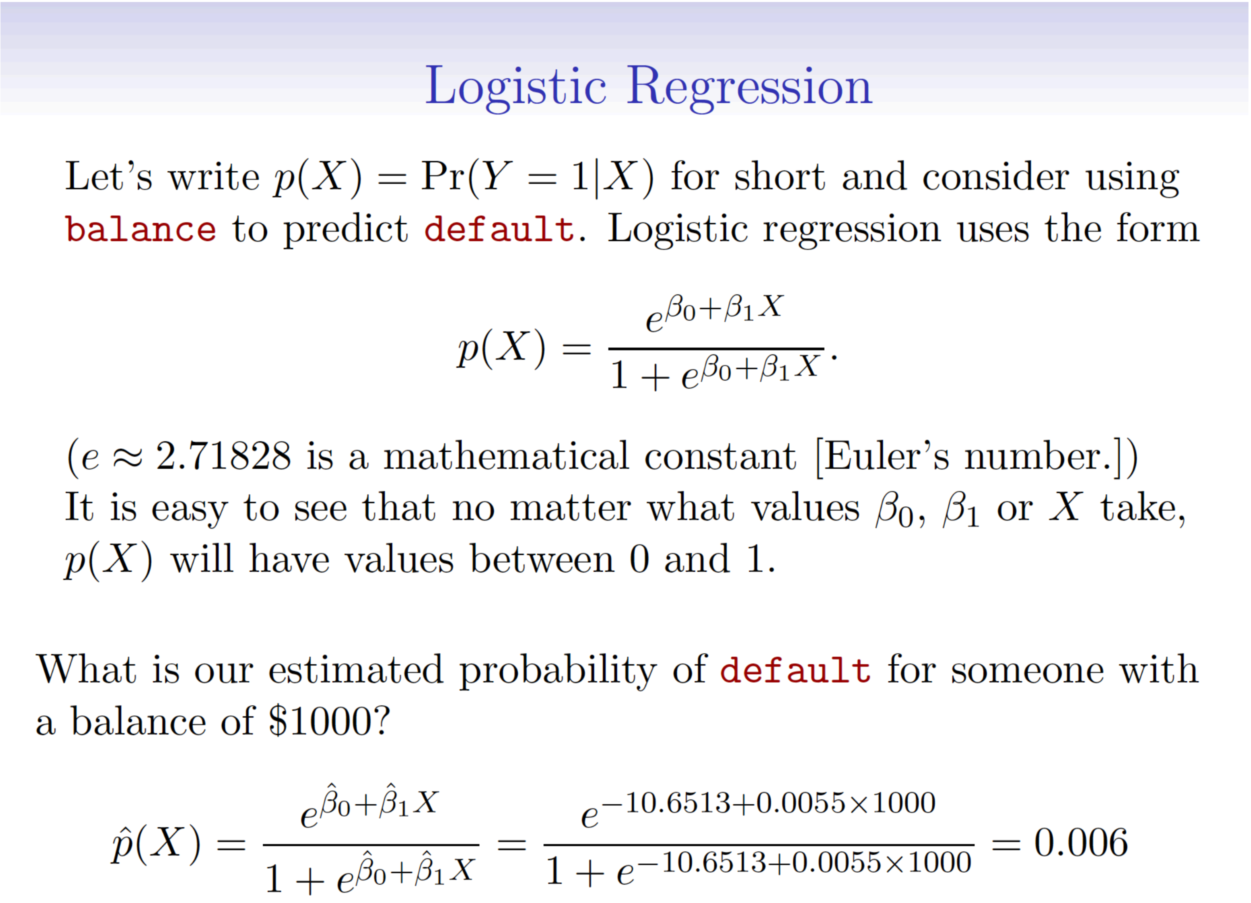
Solution:

Yes. The z-score of 24.9 for balance is very significant, about 25 standard deviations from the mean.

Q7 (5 Points) Calculate the probability of defaulting given a balance of 1000 from the stats in Q6.

Solution:

Just plug the intercept beta zero and slope beta one in to the equation below; along with an X of 1000.



## Logistic Regression

Let's write $p(X) = \Pr(Y = 1|X)$ for short and consider using balance to predict default. Logistic regression uses the form

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

($e \approx 2.71828$ is a mathematical constant [Euler's number.])
It is easy to see that no matter what values $\beta_0$, $\beta_1$ or $X$ take, $p(X)$ will have values between 0 and 1.

What is our estimated probability of default for someone with a balance of $1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1 + e^{-10.6513+0.0055\times 1000}} = 0.006$$

Q8 (5 Points)

Explain whether each scenario is a classification or regression problem.

Indicate whether we are most interested in inference or prediction.

(a) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary.

(b) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

Solution:

(a) Inference because we are interested in understanding which factors

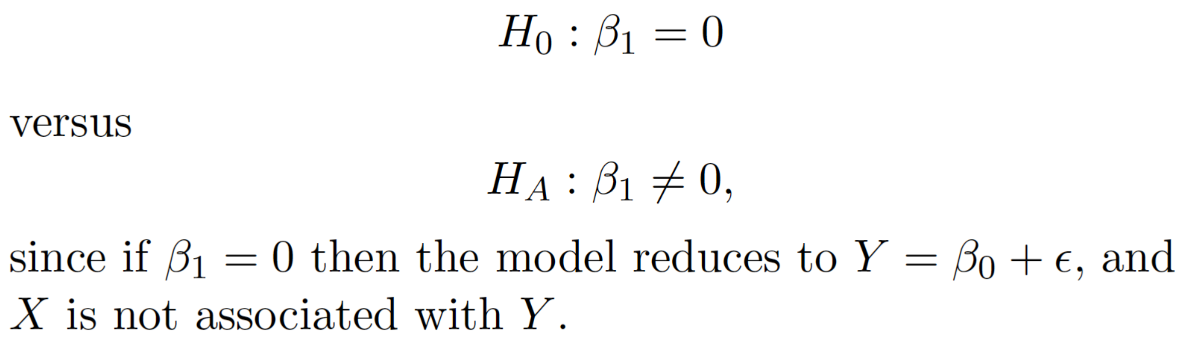
affect CEO salary. Regression because our dependent variable CEO salary is numeric.

(b) Prediction because we wish to know whether it will be a success or a failure. Classification if we predict success or a failure. Logistic regression if we predict a probability of success or a failure.

Q9 (5 Points) Describe the null hypotheses to which the p-values given in linear regression

correspond.

Solution:



$$H_0 : \beta_1 = 0$$

versus

$$H_A : \beta_1 \neq 0,$$

since if $\beta_1 = 0$ then the model reduces to $Y = \beta_0 + \epsilon$, and $X$ is not associated with $Y$.

Q10 (5 Points) What are:

- True positive rate (TPR),
- False positive rate (FPR),

How do TPR and FPR relate to AUC or AUROC?

Solution:

AUC = Area Under the Curve.
AUROC = Area Under the Receiver Operating Characteristic curve.

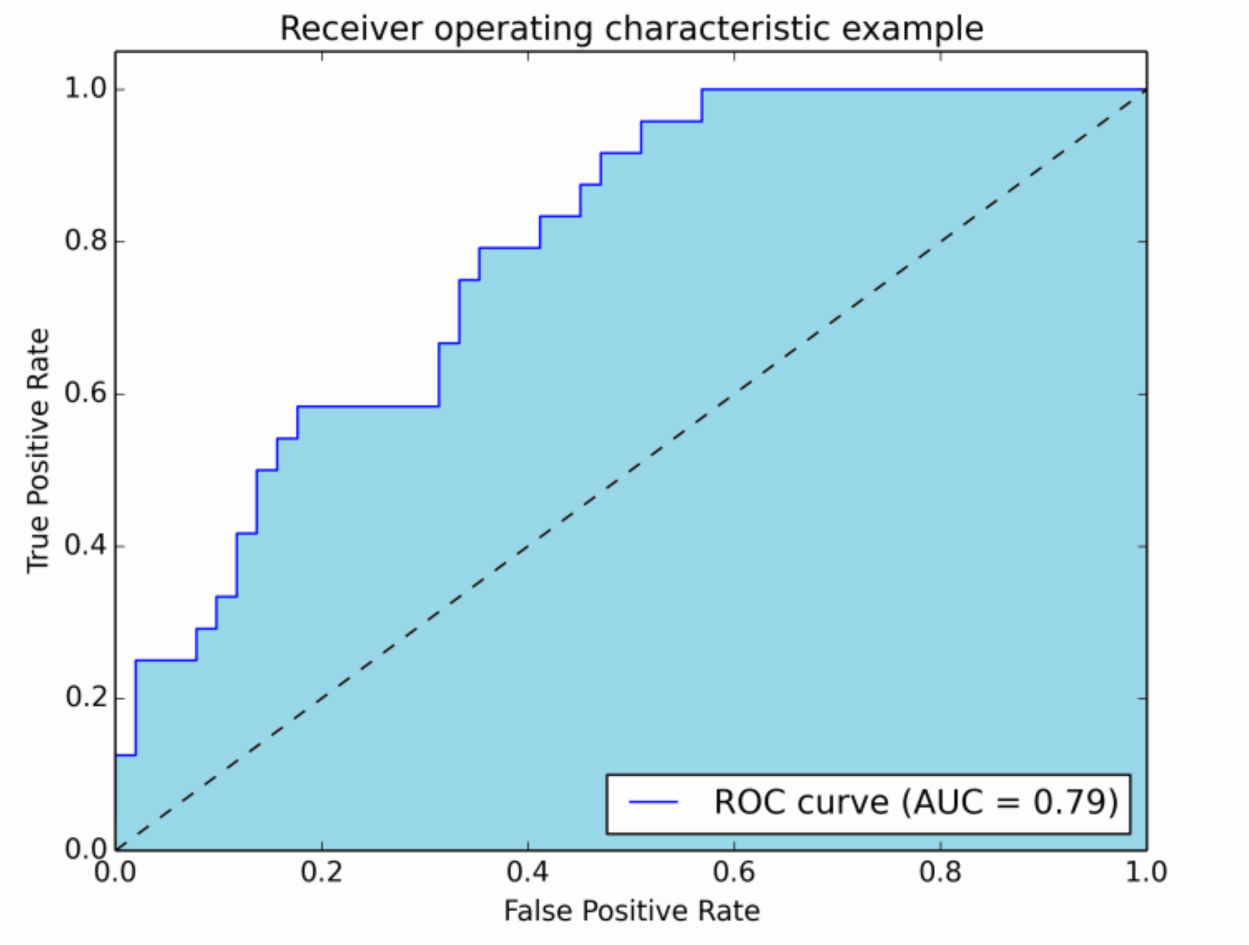
AUC is used most of the time to mean AUROC

- True positive rate (TPR), aka. sensitivity, hit rate, and recall, which is defined as TP/TP+FN. Intuitively this metric corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points. In other words, the higher TPR, the fewer positive data points we will miss.

- False positive rate (FPR), aka. fall-out, which is defined as FP/FP+TN. Intuitively this metric corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points. In other words, the higher FPR, the more negative data points will be missclassified.

To combine the FPR and the TPR into one single metric, we first compute the two former metrics with many different thresholds (for example 0.00; 0.01, 0.02,… 1.00), then plot them on a single graph, with the FPR values on the abscissa and the TPR values on the ordinate. The resulting curve is called ROC curve, and the metric we consider is the AUC of this curve, which we call AUROC.

The following figure shows the AUROC graphically:



In this figure, the blue area corresponds to the Area Under the curve of the Receiver Operating Characteristic (AUROC). The dashed line in the diagonal we present the ROC curve of a random predictor: it has an AUROC of 0.5. The random predictor is commonly used as a baseline to see whether the model is useful.

Q11 (5 Points) Bootstrap aggregating, also called bagging is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting.

Describe the algorithm in detail. What are its downsides?

Solution:

Bagging algorithm

1. Generate m new bootstrap samples

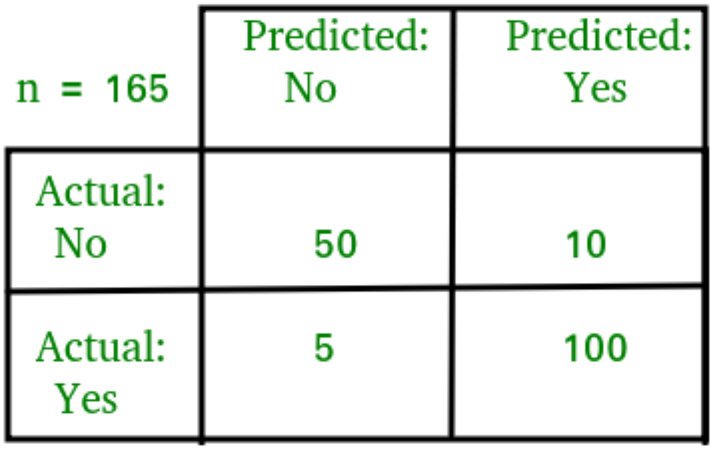
Given a standard training set D of size n, bagging generates m new training sets D_i, each of size n′, by sampling from D uniformly and with replacement.

2. Fit model on each bootstrap sample.
3. Aggregate the m predictions in an ensemble prediction.

Its downside is that one generates many, say m, bootstrap samples and fits them all. If the ensemble prediction isn’t much better than a fit on the original training set then the extra computational cost is for nothing.

See https://en.wikipedia.org/wiki/Bootstrap_aggregating

Q12 (5 Points) Given the Confusion Matrix below calculate the accuracy, recall and precision



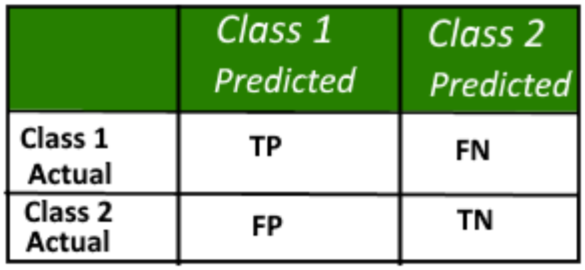
| n = 165 | Predicted: No | Predicted: Yes |
|---|---|---|
| Actual: No | 50 | 10 |
| Actual: Yes | 5 | 100 |

Solution:

Confusion Matrix see https://en.wikipedia.org/wiki/Confusion_matrix

Just use the equations below.

Confusion Matrix



| | *Class 1 Predicted* | *Class 2 Predicted* |
|---|---|---|
| **Class 1 Actual** | **TP** | **FN** |
| **Class 2 Actual** | **FP** | **TN** |

Accuracy

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Recall

$$\textbf{\textit{Recall}} = \frac{TP}{TP + FN}$$

Precision

$$\textbf{\textit{Precision}} = \frac{TP}{TP + FP}$$

Q13 (5 Points) What is a loss function? Name three cost functions used in neural networks

Solution:

See https://en.wikipedia.org/wiki/Loss_function

Quadratic cost

Also known as *mean squared error*, *maximum likelihood*, and *sum squared error*, this is defined as:

See https://en.wikipedia.org/wiki/Loss_function#Quadratic_loss_function

Cross-entropy cost

Also known as *Bernoulli negative log-likelihood* and *Binary Cross-Entropy*

See https://en.wikipedia.org/wiki/Cross_entropy

Hellinger distance

https://en.wikipedia.org/wiki/Hellinger_distance

You can find more about this here. This needs to have positive values, and ideally values between 00and 11. The same is true for the following divergences.

Kullback–Leibler divergence

Also known as *Information Divergence*, *Information Gain*, *Relative entropy*, *KLIC*, or *KL Divergence*(See here).

https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

Itakura–Saito distance

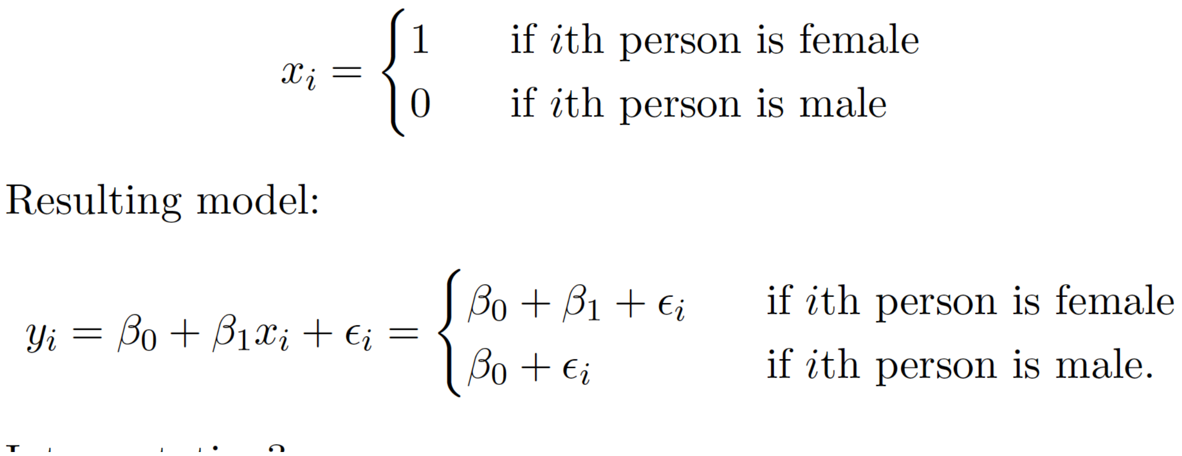
https://en.wikipedia.org/wiki/Itakura%E2%80%93Saito_distance

More loss functions at https://github.com/torch/nn/blob/master/doc/criterion.md

Q14 (5 Points) Assume one wants to use gender (male/female) as an independent variable in regression. How can we encode it?

Solution:

1 female and 0 male (or vice versa)



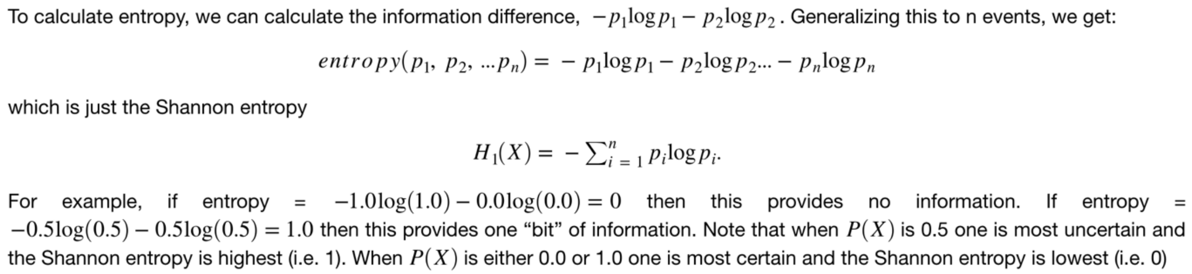
$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male} \end{cases}$$

Resulting model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i = \begin{cases} \beta_0 + \beta_1 + \epsilon_i & \text{if } i\text{th person is female} \\ \beta_0 + \epsilon_i & \text{if } i\text{th person is male.} \end{cases}$$

Q15 (5 Points) What is the equation for Shannon entropy? What is it a measure of?

Solution:



To calculate entropy, we can calculate the information difference, $-p_1\log p_1 - p_2\log p_2$. Generalizing this to n events, we get:

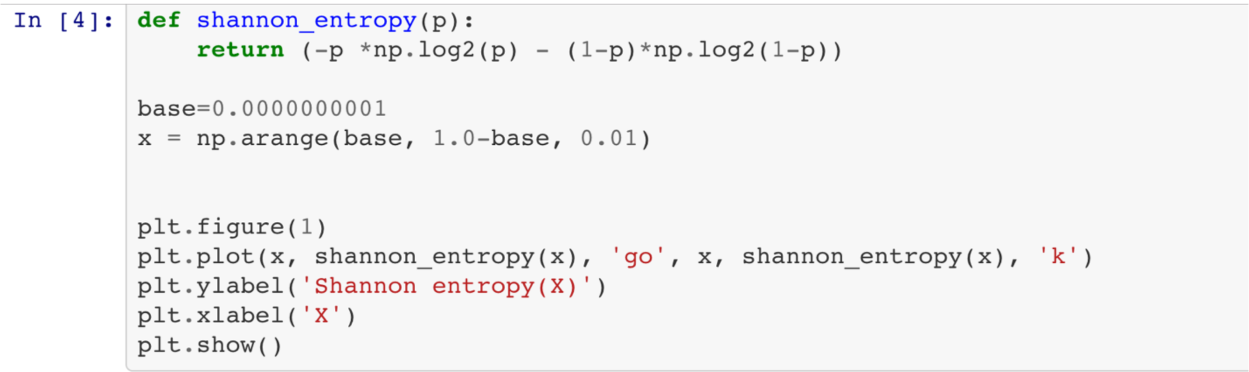
$$entropy(p_1,\ p_2,\ ...p_n) = -\ p_1\log p_1 - p_2\log p_2 ... - p_n\log p_n$$

which is just the Shannon entropy

$$H_1(X) = -\sum_{i=1}^{n} p_i \log p_i.$$

For example, if entropy = $-1.0\log(1.0) - 0.0\log(0.0) = 0$ then this provides no information. If entropy = $-0.5\log(0.5) - 0.5\log(0.5) = 1.0$ then this provides one "bit" of information. Note that when $P(X)$ is 0.5 one is most uncertain and the Shannon entropy is highest (i.e. 1). When $P(X)$ is either 0.0 or 1.0 one is most certain and the Shannon entropy is lowest (i.e. 0)

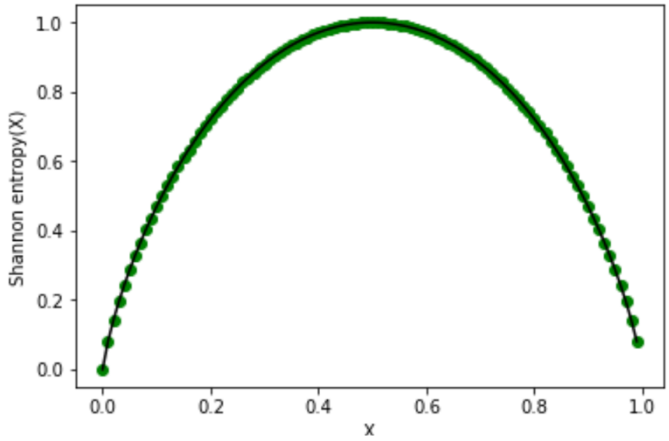
It is a measure of uncertainty/information.



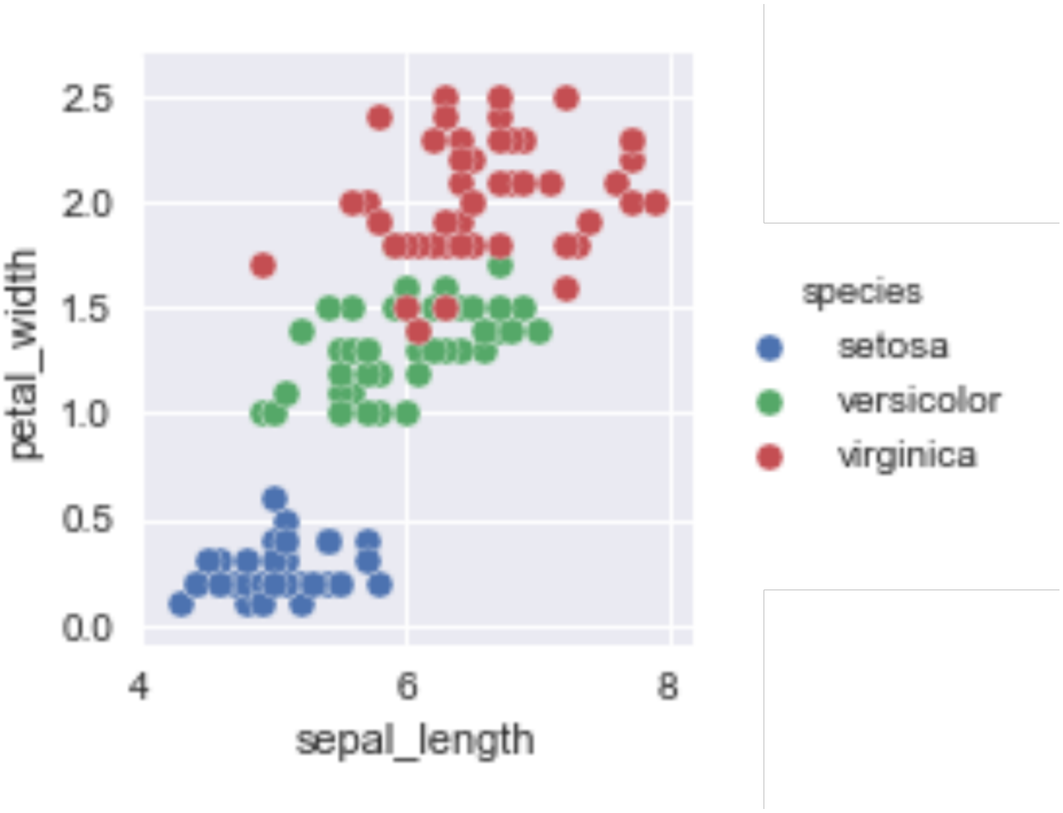
```
In [4]: def shannon_entropy(p):
            return (-p *np.log2(p) - (1-p)*np.log2(1-p))

        base=0.0000000001
        x = np.arange(base, 1.0-base, 0.01)


        plt.figure(1)
        plt.plot(x, shannon_entropy(x), 'go', x, shannon_entropy(x), 'k')
        plt.ylabel('Shannon entropy(X)')
        plt.xlabel('X')
        plt.show()
```



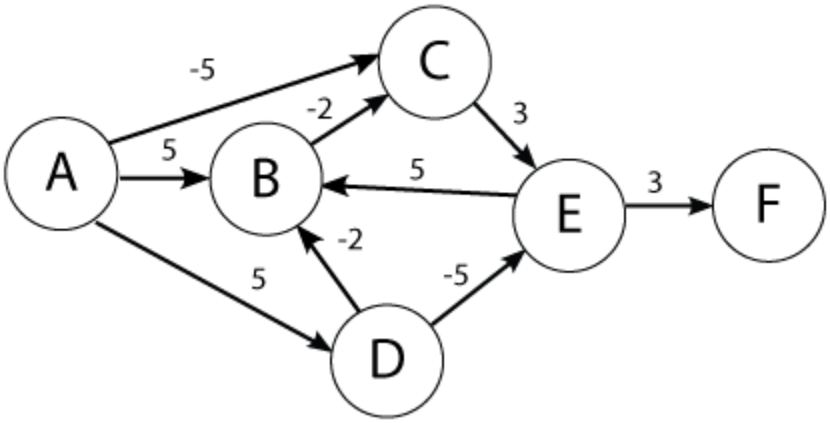
Q16 (5 Points) Below is a scatter plot of petal width versus sepal length in the famous iris data set. If you had to choose either petal width or sepal length to build a classifier for the three species of iris flowers which feature would you use? How well would it perform on this sample data?



Solution:

Clearly petal width is a better predictor. Setosa is perfectly linearly separable from the other two species with a linear discriminant at a petal width of around 0.8 and versicolor and virginica are well separated with a linear discriminant at a petal width of around 1.6.

Q17 (10 Points)



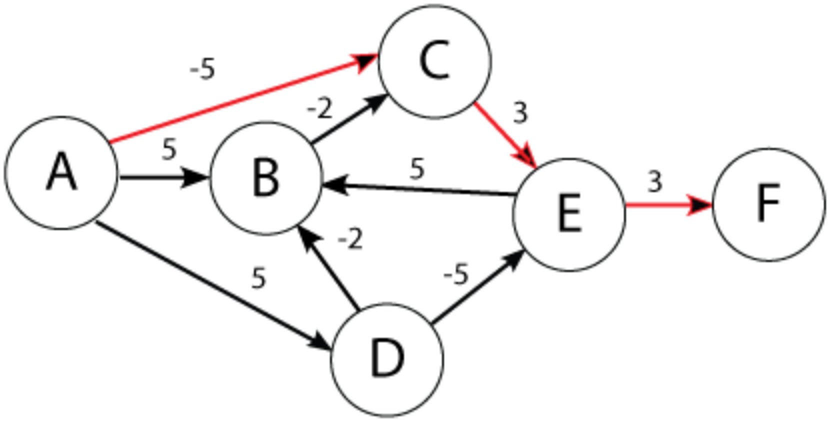
Use the Bellman-Ford algorithm to find the shortest path from node A to F in the weighted directed graph above. *Show your work.*

Solution:

Shortest path: A->C->E->F at cost 1 (-5+3+3)

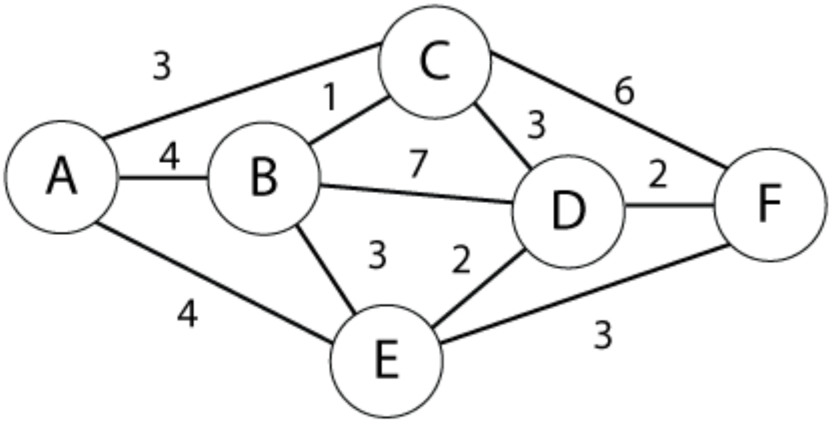
|   | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 0 | 0 | INF | INF | INF | INF | INF |
| 1 | 0 | 5 | -5 | 5 | INF | INF |
| 2 | 0 | 3 | -5 | 5 | -2 | INF |
| 3 | 0 | 3 | -5 | 5 | -2 | 1 |
| 4 | 0 | 3 | -5 | 5 | -2 | 1 |
| 5 | 0 | 3 | -5 | 5 | -2 | 1 |
| 6 | 0 | 3 | -5 | 5 | -2 | 1 |

Many students calculated from F to A which is fine but the numbers in the table will be different even though the final path will be the same.


A
B
C
D
E
F
-5
5
-2
5
3
5
-2
-5
3

Q18 (10 Points)

Find shortest path from A to F in the graph below using Dijkstra's algorithm. *Show your steps.*



Solution:

```
Given a graph, G, with edges E of the form (v1, v2) and vertices V, and a
source vertex, s

dist : array of distances from the source to each vertex
prev : array of pointers to preceding vertices
i    : loop index
F    : list of finished vertices
U    : list or heap unfinished vertices

/* Initialization: set every distance to INFINITY until we discover a path */
for i = 0 to |V| - 1
    dist[i] = INFINITY
    prev[i] = NULL
end
```

```
while(F is missing a vertex)
   pick the vertex, v, in U with the shortest path to s
   add v to F
   for each edge of v, (v1, v2)
        /* The next step is sometimes given the confusing name "relaxation"
        if(dist[v1] + length(v1, v2) < dist[v2])
```

```
            dist[v2] = dist[v1] + length(v1, v2)
            prev[v2] = v1
            possibly update U, depending on implementation
        end if
    end for
end while
```

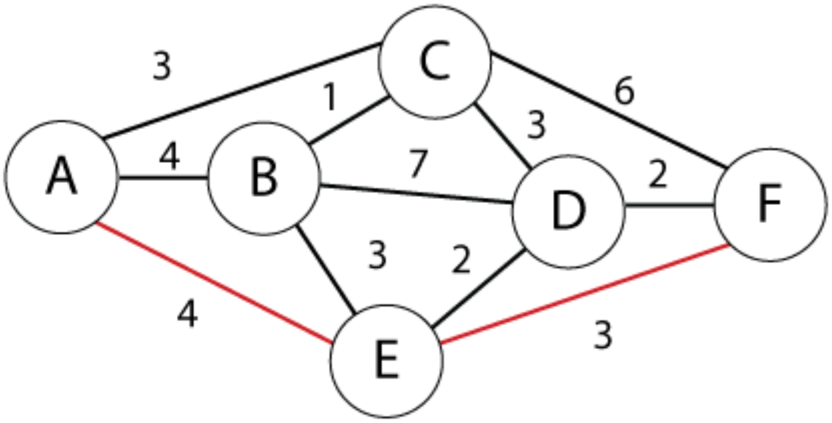
Source A

| | | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|
| 1: A {A} | A | 0 | (4,A) | (3,A)** | INF | (4, A) | INF |
| 2: C {A,C} | C | 0 | (4,A) | (3,A) | (6,C) | (4, A)** | (9,C) |
| 3: E {A,C,E} | E | 0 | (4,A) ** | (3,A) | (6,C) | (4, A) | (7,E) |
| 4: B {A,C,E,B} | B | 0 | (4,A) | (3,A) | (6,C) ** | (4, A) | (7,E) |
| 5: C {A,C,E,B,D} | D | 0 | (4,A) | (3,A) | (6,C) | (4, A) | (7,E) ** |
| 4: F {A,C,E,B,D,F} | F | 0 | (4,A) | (3,A) | (6,C) | (4, A) | (7,E) |

** U with the shortest path to s

Note: Can take in A, C, E, B, D, F or A, C, B, E, D, F as E and B both have shortest path cost 4.

Red indicates a vertex has been moved from U to F



Now we return our final shortest path, which is: A → E → F Cost (4+3 = 7)