STAT 8330: **Project 2**

Due on November 20, 2020

Cross-Enropy:

Note, I gave you the CE formula:

$$CE = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log(\hat{p}_{ik}),$$

I don't think I made it clear that you have to **encode** the y category labels to be 0 or 1. This is called **one-hot-encoding** in the machine learning world. What this means is that if y can take one of K categories, then each response has to be encoded as a K-dimensional row vector, with a 1 corresponding to the k-th element of the vector (assuming $y = k$) and zeros elsewhere. So, as an example, lets say we have $K = 3$ categories, and 4 observations, $\{3, 2, 1, 2\}$ To the one-hot-encoding of this is:

$$Y = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Now, let's assume we have predictions for each y for each class, say

$$\hat{P} = \begin{bmatrix} .3 & .2 & .5 \\ .4 & .4 & .2 \\ .8 & .2 & 0 \\ .3 & .4 & .3 \end{bmatrix}$$

So, the CE is given by (noting that the products where y is encoded as 0 are not included):

$$CE = -(log(.5) + log(.4) + log(.8) + log(.4)) = 2.75$$

A couple of notes:

- perfect classifier: $CE = 0$
- a classifier with predicted probability equals zero for the encoded class gives a negative infinite contribution in the sum. Typically, if you were calculating this, you would use a very small value instead of 0, e.g., $\log(1e - 15) = -34.53878$

Here is how you can convert your Y vector to a one-hot-encoded format:

```
> Ytrain <- read.table("Ytrain.txt", quote="\"", comment.char="")
> table(Ytrain)

Ytrain
   1    4    5    6    7    8    9   11   12
2661  430 1344 2377  550  662 1014  640  322

> library(mltools)
> library(data.table)
> Yhot <- one_hot(as.data.table(as.factor(Ytrain$V1)))
```

Here is how you can test the calculation. We will sample from a Dirichlet distribution to emulate getting classification probabilities for each category and each observaion (in this case, I am assuming each class is equally likely – this would be like flipping a 9-sided coin - your classifications should be better than this!)

```
> library(MCMCpack)
> set.seed(1)
> Ppred <- rdirichlet(10000, c(1,1,1,1,1,1,1,1,1) )
> CE = -sum(colSums(Yhot*log(Ppred)))
> CE

[1] 27122.02
```

Submission of Test Prediction File

Note, the test set, `Xtest.txt` has 5000 observations. So, you will provide a 5000 × 9 matrix of probabilities similar to `Ppred` above (but, of course, with your model's classification probabilities). Then, save that file using, e.g.,

```
> save(Ppred, file = "Ptest_groupname_mod1.RData")
```

You want to follow this exactly, and test it. I will be the one reading the files in and calculating the CE, so it needs to work.

You can submit up to 4 such matrices.