**My answers in Purple**

***Homework 3***

***Logistic Regression, Naive Bayes, and Clustering***

***Conceptual, Non-Programming***

***Naive Bayes***

* *I have a bucket with 3 red balls and 7 blue balls in it. I reach into the bucket and grab a ball. What is the proability the ball is red, P(R)? The probability that it's blue, P(B)?*

P(r ) = .3 ; P(b) = .7

* Now assume I have two buckets. Bucket 1, B1, have 5 red and 5 blue balls in it. Bucket 2, B2, has 2 red and 8 blue balls in it.
  1. I first choose a bucket and then grab a ball. Assuming I choose bucket 1 and bucket 2 with equal probability, what is the probability that the chosen ball is red? blue?

P( r ) = .35

P(b ) = .65

* 1. What is the conditional probability that I choose a red ball given B1, P(R|B1)?

P(r|B1) = P( r b1) / p(b1) = p(r )

* Let's say we're performing a binary classification problem. We're given a binary class label y and a feature vector X with dimensionality m. Meaning there are m features.
  1. What is Bayes theorem? Write it in terms of X and Y in the way it's used for our classification problem.

P(Xm | Y) = P(Xn) P(Y|Xn) / Sum(P(Xn)(P(Y|Xn), where n = (0,m))

* 1. In this problem we have 4 probabilities, {P(X), P(Y), P(X|Y), P(Y|X)}, which ones do we need to estimate?

Need to estimate every possible combination of P(Xm) & P(Y) or P(Y|Xm)

* What is the Naive Bayes assumption? Why do we need to make this assumption?

You assume all Xm or all m number of features in feature vector X ar independent of each other. If you don’t make this assumption you (or your machine learning model) must calculate all probabilities between all m features (2m).

* We discussed the following 3 different Naive Bayes implementations in SKLearn. When would you use each?
  1. Gaussian?

…when the likelihood of features is assumed to be Gaussian.

* 1. Bernoulli?

…There are multiple features but each one is assumed to be a binary-valued (Bernoulli) variable.

* 1. Multinomial?

…data is multinomially distributed.

**~~Difficult Naive Bayes~~**

~~Assume Y is boolean class label and X is a feature vector of dimension m so m is the number of features. Assume that each of the m attributes is a boolean. For example, each $x\_i$ is 0 if word i isn't present and 1 otherwise.~~

* ~~Before the Naive assumption we need to estimate P(X|Y).~~
  1. ~~How many total parameters are there?~~
  2. ~~How many parameters do we need to estimate?~~
* ~~How many parameters after making the Naive assumptions?~~

**Logistic Regression**

* What type of machine learning algorithm is logistic regression?

Supervised, regression

* In logistic regression we use the logistic function. What is the logistic function interpretted as?

It’s a sigmoid or s-shaped curve that goes from 0 to 1

* What is the assumption of logistic regression? What does the decision boundary look like?

All variables are assumed to be independent of each other.

The decision boundary is linearly fit between the two decisions (0 and 1)—if necessary, as a plane with the data transposed to a different dimension.

**~~Difficult Logistic Regression~~**

* ~~Can you show, mathematically, the decision classifier of logistic regression?~~

**Clustering**

* What type of algorithm is k-means clustering? Supervised or unsupervised?

Unsupervised.

* Does k-means clustering have any parameters? If so what are they?

Number of clusters (k)…is n\_clusters in sklearn.cluster.KMeans. also, max number of iterations, number of times algorithms will run, method for initialization, among others.

* What are the assumptions of k-means clustering?
* Assumes order in which data is clustered does not matter
* All clusters have the same sum of squared errors
* All variables have the same weight/importance from cluster to cluster

**~~More Difficult~~**

**~~Equivalence of Gaussian Naive Bayes and Logistic Regression~~**

~~Assume the following:~~

* ~~y is boolean (binary classification problem)~~
* ~~x is an m-dimensional feature vector where each feature is contiuous~~
* ~~P(X|Y) is normally distributted $P(x\_{i}|y\_{k})=\frac{1}{{\sigma\_{i}\sqrt{2\pi}}}e^{\frac{-(x-\mu\_{ik})^2}{2\sigma\_{i}^2}}$~~
* ~~Note $\sigma\_{i}$ does not depend on y but does vary by feature~~
* ~~All x's are conditionally independant given y~~

~~See if you can answer the following questions:~~

* ~~Using Bayes rule write P(Y|X). Note that this is the Gaussian Naive Bayes algorithm. The Gaussian Naive Bayes form of P(Y|X). Why?~~
* ~~Just a note: Remember that in Logistic Regression we can write $P(Y|X) = \frac{exp(w~~*~~{0} + \sum\limits~~*~~{i=1}^m w\_{i}X\_{i})}{1+exp(w~~*~~{0} + \sum\limits~~*~~{i=1}^m w\_{i}X\_{i})}$~~
* ~~Show that the Gaussian Naive Bayes form of P(Y|X) is equivalent to P(Y|X) in Logistic Regression.~~
* ~~After showing the above equivalence, what are the values of $w\_{0}$ and $w\_{i}$?~~

**~~Logistic regression MLE (Maximum Likelihood Estimation)~~**

~~In the previous question we showed that under the naive assumption Logistic Regression and Gaussian Naive Bayes are equivalent. As a result, we could estimate the weights $w\_{0}$ and $w\_{i}$ from the GNB estimates. However, we'd like a different way to estimate the Logistic Regression P(Y|X). We'd like this because we want a way to estimate that is more general than in Naive Bayes and isn't restricted by its assumptions. We can do ths via the maximum likelihood estimator, MLE. Assuming the data is iid we can write the likelihood as $\prod P(Y|X,W)$ so we want to find the weights, W, that maximize this value. Since the logarithm is a monotonically increasing function we can take the logorithm of the likelihood without changing the final answer. Taking the logarithm we get the log likelihood $l(W)= \sum\_{i=1}^{N} log(p(y\_i|x\_i, W))$. It turns out there is no closed form solution to this equation but we can used gradient based methods to find the maximum/minimum~~[~~gradient methods~~](https://en.wikipedia.org/wiki/Gradient_descent)~~.~~

* ~~Find the gradient of the log-likelihood function~~
* ~~Read about the gradient descent/ascent algorithm~~
* ~~What is the update rule for gradient ascent in the case of the logistic regression log-likelihood?~~
* ~~Add a regularization term to the log-likelihood function so it becomes $\sum\_{i=1}^{N} log(p(y\_i|x\_i, W)) + \frac{\lambda}{2}||W||^2$. Note the similarity with the regularized ridge and lasso regression from before. The divide by 2 helps with cancelling later on when differentiating.~~
* ~~Recalculate the gradient of the new objective function. What is it?~~
* ~~What is the new update rule for the gradient based optimization~~
* ~~Try to implement this in Python for logistic regression so you can calculate the MLE estimates of the weights.~~