MTH 511a - Mini Project

Instructor: Dootika Vats

Due: 13th November at 8:00pm

Please read the instructions on submission **very** carefully. The grading will be **automatic**, so if you do not follow proper directions, you will not be able to get any marks. This mini-project counts for 10% of the overall grade.

Each and every one of you is provided with a dataset of 50 covariates (including intercept) and a binary response y. There are 1000 observations. You can load the data in R using the following command:

```
dat <- read.csv("https://dvats.github.io/assets/data/rollnumber.csv")
```

where replace `rollnumber` with your roll number.

The goal in this assignment is to minimize *misclassification loss* for a new independent dataset (that only I have). You are to submit your final model in the form of a function, and I will calculate the average misclassification on this independent dataset.

For some mathematical context, the data consists of $Y_i \sim \text{Bern}(p_i)$, where $p_i = g(X_i, \beta)$ for some function g. This defines a log-likelihood:

$$l(\beta|y, X) = \sum_{i=1}^{n} \log f(y_i|\beta, X_i)\,.$$

Depending on your choice of p_i, you can obtain maximum likelihood estimates of β. You may also choose to add penlization by adding ridge or bridge penalty to the negative log-likelihood. (I highly recommend calculating the Hessian and implementing a Newton-Raphson algorithm if feasible).

You are allowed to fit any model for estimation of the response. This includes, logistic regression, penalized logistic regression, probit regression (from the mid-sem exam), and penalized probit regression. *I will not know what model you fit*!

Once you're ready with your model and final estimates of β, save your regression estimates in a column matrix `beta`.

Write a function `est.y` that has arguments `X` (model matrix) and `beta` (regression estimates). In this function you will calculate the estimated values of the response y for a given input matrix `X.new`. I will input `X.new` matrix of size $n_1 \times 50$ for some $n_1 > 0$ and the function should output `y.pred` a $n_1 \times 1$ column vector.

```
est.y <- function(X.new, beta)
{
  y.pred <- ...
  return(y.pred)
}
```

(The line `y.pred` comes from the function g that you've chosen.)

This is important: At the end of your script, run the following code:

```
save(est.y, beta, file = "rollnumer.Rdata")
```

This will save your function `est.y` and `beta` in a file with your roll number as the name, in your current working directory. (To see your working directory, type `getwd()` in the `R` console).

When done, please upload your `rollnumber.Rdata` file in the following Dropbbox link:

`https://www.dropbox.com/request/2bVnhbHZQ0t6a4VeqYm3`

When submitting, please follow the instructions:

- **Please sign out** of Dropbox if you are signed in to Dropbox.
- Under "Your Name" write down your **roll number**
- Under "Your Email" use your **iitk email id**.

Example

Below is an example code. The code implements linear regression (continuous response) for the `cars` dataset.

```
data(cars)
y <- cars$dist
X <- cbind(1, cars$speed)

# regression estimate
# one can try other penalization methods and
# choose which one works best.
```

```r
beta <- solve(t(X) %*% X)%*% t(X) %*% y

# estimated response for new X
# for linear regression, the details
# of this function will be different
# for logistic regression
est.y <- function(X.new, beta)
{
  y.pred <- X.new %*% beta
  return(y.pred)
}
```

Some words of caution

- Make sure your optimization routines don't consume all `max.iter`. If this happens, then this means you haven't converged as yet to the estimator.
- One way to check whether you've reached a local optima is to calculate the gradient vector at the kth iteration and check whether it's close to zero: $\|\nabla f(\theta_{(k)}\| \approx 0$
- Follow the naming conventions I use here, otherwise you may not get any points. Particularly, make sure your final estimates are stored in `beta` and your function name is `est.y`. Do not use any other names for these two objects!
- Before submitting the `rollnumber.Rdata` file, make sure everything works. Change your working directory to the folder that contains `rollnumber.Rdata`. Run the following lines

  ```r
  rm(list = ls())
  load("rollnumber.Rdata")
  ```

  This will first clear all memory of the R session and then load your `.Rdata` file. After this, call `est.y(X, beta)` using a dummy X matrix you create of size $n_1 \times 50$, for any n_1. If you get back a column vector of length n_1 of 1s and 0s, then that means your function will give me no errors.

Good luck and SUBMIT ON TIME!