Stat 435
Arnav Palkhiwala

1. Summary:

This week we further delved into Linear Regression focusing on multiple regression where you have more than one predictor. We also looked at conducting F Tests when evaluating models with different numbers of predictors and finding the best models and the predictors that are statistically significant and useful for the model. We then went into classification and understanding logistic regression and the logistic regression function when you are picking between two classes. We also look at the odds ratio and use maximum likelihood estimation and gradient descent in the classification setting.

2. Concepts:

Multiple Linear Regression:

- The first goal of MLR is to estimate each of the beta coefficients
- A useful trick: Y - E[Y] to center the data so you don't need to deal with beta 0
- If you have X be an n by p matrix, Y = X Beta + Epsilon when Beta is a vector of betas
- To estimate Beta coefficients, we use ordinary least squares and you want to minimize the RSS
- argmin beta (||Y - XB||2), an n dimensional vector containing residuals
- L2 norm notation is to reduce the amount of notations to describe RSS
- You want to take the gradient of above and set it equal to 0 and you get Beta hat = (X^T*X)^(-1)X^TY
- The matrix X has to be invertible so n has to be >= p
- For inference, beta hat ~ N(Beta, sigma^2(XTX)^(-1))
- Then you have a t distribution (Beta hat - Beta)/SE(Beta hat i) which is T(n-p-1)
- SE(Beta) = sqrt(RSS/(n-1)
- F observed = ((TSS - RSS)/P)/(RSS/(n-p-1)) which is an F distribution (p, n - p - 1)
- So you compute F observed and the p value
- You can also compute confidence interval bc you know its a t distribution
- You can use Bonferroni correction for the multiplicity effect

Classification:

- We cannot perform regression in classification settings because you would not have a probability value between 0 and 1 because regression could produce values outside of this range
- In logistic regression, you have one predictor with two classes of Y to predict from
- Goal is approximate P(Y = 1 | X) and it is 1 when that probability is > ½ or else 0

- C(x) = argmin any classifier g (P(Y != g(x))
- One approach: q(x) = Beta 0 + Beta1 * x
- Best, Logistic Model is q(x) = 1/(1 + e^-(Beta0 + Beta1 x))
- If Beta0 + Beta1 x is large, q(x) is about 1 vs 0
- Odds ratio is P(Y = 1 | X)/P(Y = 0 | X) which is e^(Beta0 + Beta1 x)
- Log of odds ratio is Beta0 + Beta1 x
- We can then use Maximum Likelihood Estimator
- This is since our data is I.I.D
- After using MLE and then gradient descent we can easily find the optimal solution

3. Uncertainties:

I am curious about how classification will work with when you have more than one class because you cannot just see if the probability is greater than ½ which is the threshold for 2 classes. Would you want to just find the class that has the maximum probability but I am not sure now we would go about that. I also want to look more into gradient descent and MLE and how exactly we are using it in this situation and how it can help with the maximizing needed in classification. I want to understand exactly how this function works and I have heard about the sigmoid function which seems to be the logistic regression function. So, I want to learn more about this and understand this relationship better.