The University of Texas at Austin
Data Science Lab — Fall 2023

Lab Two

Alex Dimakis Due: Monday, September 11, Midnight.

Problem 1: Covariance.

- When given a data matrix, an easy way to tell if any two columns are correlated is to look at a scatter plot of each column against each other column. For a warm up, do this: Look at the data in `DF1` in `Lab2_Data.zip`. Which columns are (pairwise) correlated? Figure out how to do this with Pandas, and also how to do this with Seaborn.
- Compute the covariance matrix of the data. Write the explicit expression for what this is, and then use any command you like (e.g., np.cov) to compute the 4×4 matrix. Explain why the numbers that you get fit with the plots you got.
- The above problem in reverse. Generate a zero-mean multivariate Gaussian random variable in 3 dimensions, $Z = (X_1, X_2, X_3)$ so that (X_1, X_2) and (X_1, X_3) are uncorrelated, but (X_2, X_3) are correlated. Specifically: choose a covariance matrix that has the above correlations structure, and write this down. Then find a way to generate samples from this Gaussian. Choose one of the non-zero covariance terms (C_{ij}, if C denotes your covariance matrix) and plot it vs the estimated covariance term, as the number of samples you use scales. The goal is to get a visual representation of how the empirical covariance converges to the true (or family) covariance.

Problem 2: Outliers.
Consider the two-dimensional data in `DF2` in `Lab2_Data.zip`. Look at a scatter plot of the data. It contains two points that look like potential outliers. Which one is "more" outlying? Propose a transformation of the data that makes it clear that the point at $(-1, 1)$ is more outlying than the point at $(5.5, 5)$, even though the latter point is"farther away" from the nearest points. Plot the data again after performing this transformation. Provide discussion as appropriate to justify your choice of transformation. *Hint: if* $\mathbf{y}$ *comes from a standard Gaussian in two dimensions (i.e., with covariance equal to the two by two identity matrix), and*

$$Q = \begin{pmatrix} 2 & \frac{1}{2} \\ \frac{1}{2} & 2 \end{pmatrix},$$

what is the covariance matrix of the random variable $\mathbf{z} = Q\mathbf{y}$*? If you are given* $\mathbf{z}$*, how would you create a random Gaussian vector with covariance equal to the identity, using* $\mathbf{z}$*?*

Problem 3: Popular Names.
The goal of this exercise is for you to get more experience with Pandas, and to get a chance to explore a cool data set. Download the file `Names.zip` from Canvas. This contains the frequency of all names that appeared more than 5 times on a social security application from 1880 through 2015.

- Write a program that on input k and $XXXX$, returns the top k names from year $XXXX$ starting with the letter "s".

- Write a program that on input `Name` returns the frequency for men and women of the name `Name`. Also find the most common first letter in names for men and women respectively across all years.
- It could be that names are more diverse now than they were in 1880, so that a name may be relatively the most popular, though its frequency may have been decreasing over the years. Modify the above to return the relative frequency. Note that in the next coming lectures we will learn how to quantify diversity using entropy.
- Find all the names that used to be more popular for one gender, but then became more popular for another gender.
- For a given year $YYYY$, identify the name with the highest surge in popularity compared to the previous year. Define "surge" as the largest percentage increase in frequency.
- (Optional) For a given range of years, say from $YYYY_1$ to $YYYY_2$, determine the top 3 names that have consistently increased in popularity during this period.
- (Optional) Find something cool about this data set.

Problem 4: Starting in Kaggle.
Later in this class, you will be participating in the in-class Kaggle competition made specifically for this class. In that one, you will be participating on your own. This is a warmup- the more effort and research you put into this assignment the easier it will be to compete into the real Kaggle competition that you will need to do soon.

1. Let's start with our first Kaggle submission in a playground regression competition. Make an account to Kaggle and find
   https://www.kaggle.com/c/house-prices-advanced-regression-techniques/
2. Follow the data preprocessing steps from
   https://www.kaggle.com/apapiu/house-prices-advanced-regression-techniques/regularized-linear-models. Then run a ridge regression using $\alpha = 0.1$. Make a submission of this prediction, what is the RMSE you get?
   (Hint: Remember to exponentiate `np.expm1(ypred)` your predictions).
3. Compare a ridge regression and a lasso regression model. Optimize the alphas using cross validation. What is the best score you can get from a single ridge regression model and from a single lasso model?
4. Plot the l_0 norm (number of nonzeros) of the coefficients that lasso produces as you vary the strength of regularization parameter alpha.
5. Add the outputs of your models as features and train a ridge regression on all the features plus the model outputs (This is called Ensembling and Stacking). Be careful not to overfit. What score can you get? (We will be discussing ensembling more, later in the class, but you can start playing with it now).
6. Install XGBoost (Gradient Boosting) and train a gradient boosting regression. What score can you get just from a single XGB? (you will need to optimize over its parameters). We will discuss boosting and gradient boosting in more detail later. XGB is a great friend to all good Kagglers!

7. Do your best to get the more accurate model. Try feature engineering and stacking many models. You are allowed to use any public tool in python. No non-python tools allowed.

8. (Optional) Read the Kaggle forums, tutorials and Kernels in this competition. This is an excellent way to learn. Include in your report if you find something in the forums you like, or if you made your own post or code post, especially if other Kagglers liked or used it afterwards.

9. Be sure to read and learn the rules of Kaggle! No sharing of code or data outside the Kaggle forums. Every student should have their own individual Kaggle account and teams can be formed in the Kaggle submissions with partners. This is more important for live competitions of course.

10. As in the real in-class Kaggle competition (which will be next), you will be graded based on your public score (include that in your report) and also on the creativity of your solution. In your report (**that you will submit as a pdf file**), explain what worked and what did not work. Many creative things will not work, but you will get partial credit for developing them. We will invite teams with interesting solutions to present them in class.