CS 189 Introduction to Machine Learning
Fall 2018

HW8

1 Getting Started

Read through this page carefully. You may typeset your homework in latex or submit neatly handwritten/scanned solutions. Please start each question on a new page. Deliverables:

1. Submit a PDF of your writeup to assignment on Gradescope, “HW8 Write-Up”. If there are graphs, include those graphs in the correct sections. Do not simply reference your appendix.

(a) Who else did you work with on this homework? In case of course events, just describe the group. How did you work on this homework? Any comments about the homework?

(b) Please copy the following statement and sign next to it. We just want to make it *extra* clear so that no one inadverdently cheats.

I certify that all solutions are entirely in my words and that I have not looked at another student’s solutions. I have credited all external sources in this write up.

This homework is due **Wednesday, November 21 at 10pm.**

2 Fairness in Practice: Criminal Justice Case Study

Risk assessment is an important component of the criminal justice system. In the United States, judges set bail and decide pre-trial detention based on their assessment of the risk that a released defendant would fail to appear at trial or cause harm to the public. While *actuarial risk assessment* is not new in this domain, there is increasing support for the use of learned risk scores to guide human judges in their decisions. Proponents argue that machine learning could lead to greater efficiency and less biased decisions compared with human judgment. Critical voices raise the concern that such scores can perpetuate inequalities found in historical data, and systematically harm historically disadvantaged groups.

In this problem, we'll begin to scratch at the surface of the complex criminal justice domain. Our starting point is an investigation carried out by ProPublica[1] of a proprietary risk score, called COMPAS score. These scores are intended to assess the risk that a defendant will re-offend, a task often called recidivism prediction. Within the academic community, the ProPublica article drew much attention to the trade-off between separation and sufficiency that we saw earlier.

We'll use data obtained and released by ProPublica as a result of a public records request in Broward Country, Florida, concerning the COMPAS recidivism prediction system [2]. The data is available at `https://raw.githubusercontent.com/propublica/compas-analysis/master/compas-scores-two-years.csv`. Following ProPublica's analysis, we'll filter out rows where `days_b_screening_arrest` is over 30 or under -30, leaving us with $6,172$ rows (you'll need to download the data from the link above and apply this filter).[3]

(a) Calibration/sufficiency

(i) Plot the fraction of defendants recidivating within two years `(two_year_recid == 1)` as a function of risk score `(decile_score)`, for black defendants `(race == "African-American")` and white defendants `(race == "Caucasian")`.

When comparing two line plots, it is often useful to have confidence intervals around the points you are comparing. Since recidivism rate is a score between 0 and 1, we might use confidence intervals for a binomial random variable, as in `statsmodels.stats.proportion.proportion_confint()`. Whether you decide to visualize confidence intervals or not, explain your choice.

(ii) Based on these plots, does the risk score satisfy sufficiency across racial groups in this dataset? This is somewhat subjective, since we want to allow for approximate equality between groups; justify your answer in a sentence or two.

[1] Julia Angwin et al., "Machine Bias," ProPublica, May 2016, `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`.

[2] `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`

[3] The pandas python library (`https://pandas.pydata.org/`) can be very helpful for handling operations on CSV-like datasets. We suggest you start there, using `pandas.read_csv()` to load in the data. For reporting results, you might find `pandas.DataFrame.to_latex()` helpful.

(b) Error rates/separation

   (i) Plot the distribution of scores received by the positive class (recidivists) and the distribution of scores received by the negative class (non-recidivists) for Black defendants and for White defendants.

   (ii) Based on these plots, does COMPAS achieve separation between the risk score and race?

   (iii) Report the Positive Predictive Value, False Positive Rate, and False Negative Rate for a risk threshold of 4 (i.e., defendants with `decile_score >= 4` are classified as high risk), for Black defendants and for White defendants.

   (iv) Can we pick two thresholds (one for black defendants, one for white defendants) such that FPR and FNR are roughly equal for the two groups (say, within 1% of each other)? What is the PPV for the two groups in this case? Note: trivial thresholds of 0 or 11 don't count.

(c) Risk factors and interventions
Report the recidivism rate of defendants aged 25 or lower, and defendants aged 50 or higher. Note the stark difference between the two: younger defendants are far more likely to recidivate.

Note: you get to make some design decisions about how you produce your tests and interpret results in this problem. It's meant to be open ended, but please state and justify your decisions (e.g. confidence parameters on any confidence intervals), as well as give ample but concise justifications for any conclusions you reach from your analysis.

3 Data Modeling of Traffic Stops

For this problem we'll use data released by the Stanford Open Policing Project (SOPP) for the state of North Carolina. It contains records of 9.6 million police stops in the state between 2000 and 2015.

(a) We've merged these datasets and provided for you a data frame of population and stop counts ("stop_counts_df_NC.csv") for each possible group defined by race, age, gender, location, and year, where:

- race is one of "Asian", "Black", "Hispanic", "White"
- age is one of the buckets 15–19, 20–29, 30–39, 40–49, and $50+$
- gender is one of "female", "male"
- location is a state patrol troop district
- year is between 2010 and 2015, inclusive
- Population is the recorded population of this group according the census data, and
- Count is the number of reported stops for this group according to the SOPP data

Use this data to get the per-capita stop rates of each race group. Do the same thing for driver gender, and report the results in tables below. (hint: the `pandas.groupby()` function might be helpful)

(b) Denote variables $G \in \{m, f\}$ for gender group, $A \in 1, 2, 3, 4, 5$ for each of the five age groups defined above, and $Y \in \{0, 1\}$ for the outcome of being stopped or not. Suppose that $Y \perp G|A$, so that knowing an individual's age group makes knowing their gender irrelevant in the probability they will be stopped. **Use the law of total probability to find precise conditions such that in this setting, $\mathbb{P}[Y|G = m] \neq \mathbb{P}[Y|G = f]$.** (hint: your conditions should be on the equality or inequality of conditional probability distributions).
What does this say about our ability to discern in stop rates for different race groups and genders through analysis of the tables generated in the previous part?

(c) For the remaining parts we want to analyze the relationship between race and stop frequency. In light of the previous part, we will fit a negative binomial regression as given in page 5 of the SOPP paper[4]. Using the subscript $[ragly]$ to denote a specific instantiation of race, age, gender, location, and year, we fit model parameters $\mu, \alpha_r, \beta_a, \gamma_g, \delta_l, \epsilon_y$

$$y_{ragly} \sim NB\left(n_{ragly}e^{\mu+\alpha_r+\beta_a+\gamma_g+\delta_l+\epsilon_y}, \phi\right)$$

Where α_r represents the coefficient for race r, β_a the coefficient for age group a, and so on (μ is an offset parameter and ϕ is an overdispersion parameter allowing for variation in y due to factors other than the five we consider). Since we are estimating counts, we use n_{ragly} to account for the total population in each race-age-gender-location-year group $[ragly]$. Note that allowing for specific coefficients for factors like age, gender, location, and year in the equation above helps us disambiguate their effects on the overall stop rates in each $[ragly]$ group separately from the effect of race, so that we say such a model "controls" for these effects.

Fit a negative binomial model to the full dataset in part (a)[5]. (The "Population" column in your data corresponds to the "exposure" variable in most frameworks. Equivalently, "offset" is the log of the exposure.)

Report the coefficients of race, age, and gender. Based on these coefficients, what is the ratio of stop rates of Hispanic drivers to White drivers, and Black drivers to White drivers, controlling for age, gender, location, and year? (*hint: note that the coefficients are given in the exponent form.*)

(d) Give three distinct potential reasons for the racial disparity in stop rate as measured in part (c).

(e) Now we want to investigate the difference in post-stop outcomes for each race. The corresponding data is in "stop_outcome_df_NC.csv". Since we're investigating the effect of many variables on a rate between 0 and 1, a logical choice of model is logistic regression. If p_{ragly}

[4] `https://5harad.com/papers/traffic-stops.pdf`

[5] There are many existing implementations available to help you do this, we recommend `statsmodels.api.NegativeBinomial.from_formula()`.

denotes the proportion of the group described by a specific instantiation of race, age, gender, location, and year (denoted again using the subscript $[ragly]$), logistic regression fits the following relation:

$$p_{ragly} \approx \frac{1}{1+e^{-\left(\mu+\alpha_r+\beta_a+\gamma_g+\delta_l+\epsilon_y\right)}}$$

by optimizing over the parameters on the right hand side of the equation above.

(i) Controlling for age (as in parts A, B), gender, year, and location, use logistic regression[6] to estimate impact of race on:

- probability of a search `(search_conducted)`
- probability of arrest `(is_arrested)`,
- probability of a citation `(stop_outcome == "Citation")`

(ii) For each of the three outcomes, report the coefficients of race, age, and gender along with standard errors of those coefficients. Feel free to sample the data for performance reasons, but if you do, make sure that all standard errors are < 0.1.

(f) Interpret the coefficients you reported in part (e). Interpreting the coefficients is slightly subjective, and will likely require approximations; be specific and justify how you are performing computations, and for which of the three outcomes the approximations are most likely to hold. (*hint: for small values of* $\frac{1}{1+e^{-x}}$, $\frac{1}{1+e^{-x}} \approx e^x$.)

- What is the ratio of the probability of search of Hispanic drivers to White drivers? Black drivers to White drivers?

- Repeat the above for the probability of arrest instead of search.

- What is the difference in citation probability between Hispanic drivers and White drivers? Black drivers and White drivers?

- Comment on the age and gender coefficients in the regressions.

(g) In part (b), we explained one reason why we might want to control for variables such as gender and location in the regression, and why the results might not be what we want if we don't control for them. However, decisions about what to control are somewhat subjective. **What is one reason we might *not* want to control for location in testing for discrimination?** In other words, how might we underestimate discrimination if we control for location? (Hint: broaden the idea of discrimination from individual officers to the systemic aspects of policing.)

[6] There are many existing implementations available to help you do this, one is recommend `statsmodels.api.logit()`.

(h) The SOPP authors provide a README[7] file in which they note the incompleteness, errors, and missing values in the data on a state-by-state level. Pick any two items from this list and briefly explain how each could lead to errors or biases in the analyses you performed (or in the other analyses performed in the paper).

4 Predicting Accessibility of Buildings

In this question, we will build and test a classifier that takes in the image of a building entrance, and predicts whether the building is accessible to an individual in a wheelchair, or not. We will do this using the image data that you collected in a previous homework, which we've compiled and provided for you in the following files:

- "cs189_image_featues.npz" contains the featurized image data
- "cs189_image_metadata.pkl" contains the metatata associated with each image, where the "id" field corresponds to the row number in "cs189_image_featues.npz"
- "cs189_images_accessibility.zip" contains the image png files. You can download it from `https://s3-us-west-2.amazonaws.com/189images/cs189_images_accessibili` `zip`, but note that this is a big file so download it only if you want to use the images for part (f).

For this problem we extracted features from our roughly 15,000 images by extracting the last layer from a pretrained residual network (the network was trained on 1.2 million images of objects containing various categories). We will use the activations from the last layer as our feature embedding for this problem. Note we are not training a neural network for this task but simply using the activations of a previously trained network (pre-trained network). The exact model used for the feature extraction can be found here: `https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py`.

For all of your plots: label axes, title the plots, and provide legends when applicable.

(a) Load the data and metadata from the files above. You'll note that there are some submissions for which there is no latitude or longitude information, so we'll have to 'clean' the data a bit. (Note that we've heavily cleaned up many submissions for you to get it into this format already). Specifically, **remove from the dataset any entries for which the lat or lon entries in the metadata dictionaries cannot be cast to a float. Report how many such entries you had to remove.**

(b) Split the data into a validation set corresponding to 3000 randomly chosen points, and put the remaining data in the training set. Configure the y labels so that an accessible building has a label of 1, and a non-accessible building has a label of -1. (The labels provided are 1 for accessible and 0 for inaccessible).

[7] `https://github.com/5harad/openpolicing/blob/master/DATA-README.md`

(c) Fit a ridge regression predictor

$$\hat{w} = \arg\min_{w \in \mathbb{R}^d} \|X_{train}w - y_{train}\|_2^2 + \lambda\|w\|_2^2$$

. We'll investigate models of the form

$$\hat{y}(x) = \begin{cases} 1 & if\ \hat{w}^\top x \geq t \\ -1 & if\ \hat{w}^\top x < t \end{cases}$$

for some threshold t.

Start with regularization parameter $\lambda = 1e^{-4}$ and $t = 0$. Train this predictor on the training data. Next, apply the predictor $\hat{y}$ to both the train set of roughly 12,000 points, and the validation set of 3,000 points.

- **Plot and interpret the ROC curves. What is the TPR and FPR on the test set resulting from a classification threshold of 0?**
- **Also report the accuracy from `sklearn.metrics.accuracy_score`. What metric does this function report?**

(d) Now we'll investigate how the number of training datapoints affects the performance of our model on a holdout set. **Plot performance on a *fixed* holdout set of size $3,000$ points, as you vary the number of training datapoints.** Plot results for at least the following number of training datapoints: $[1000, 2000, 4000, 8000, 12000]$. **What do you notice?**

(e) Along with the labels $\{$inaccessible, accessible$\}$, we collected the latitude and longitude of each data point. Now, we will investigate if our model is making mistakes in certain regions across campus. **First , write down a hypothesis as to how errors will be spread geographically across campus.**

Next, we will visualize errors across campus for the test set. You can do this in `matplotlib.pyplot.scatter` by setting input data as the latitude and longitude of points in the test set, the `c` parameter to be the absolute error associated with each point. Use the bounds $lat \in [37.87, 37.88]$, $lon \in [-122.28, -122.25]$. You may want to also plot the following points of interest to give yourself a notion of scale and extent of this region (this is optional). Specifically:

- Soda Hall is at approximately $[37.875587, -122.258352]$.
- The center of memorial glade is at approximately $[37.873118, -122.259441]$.

Provide the vizualization with these bounds in your writeup. Do errors seem to exhibit patterns over space? Does this match your hypothesis?

(f) Devise a question and hypothesis of your own, and write code/visualizations to test this hypothesis. Clearly document the hypothesis, testing process, analysis, and conclusions. There should be enough detail that one of your peers could reproduce your analysis. If you need help getting started, some example ideas are

- Investigate accuracies for different submission IDs (each sumbission id is a chunk of $40-80$ images). What does the distribution of average errors look like across submissions? Are there chunks of submissions that have conspicously high/low number of errors?
- Redo the plots from part (d) using 5-fold cross validation to get error bars around each point (note that you will still want to keep each of the 5 validation sets a constant size across different train test sizes and nubmer of features). How does the variance in accuracy seem to scale with increased training set size?
- Look into the images themselves; is there a pattern in images that were mis-predicted?

but we encourage you to develop your own ideas!

This is meant to be fun, and not overly laborious, so pick a question that is both interesting to you and is testable from the data provided. The main point of this question is to do the following: (a) pose your own formal hypothesis, (b) create the functionality to test it, (c) perform your own analysis, and (d) report on your findings with respect to your original hypothesis. Whatever you choose to investigate, please be complete in your writeup.

(g) Having conducted the analysis in the previous parts, or from your own experience collecting images, give three concrete ways we could change or augment the image collection and labelling instructions from the previous homework in order to get more standardized data.