Go to piazza and open today's lecture notes in the hub!

https://piazza.com/class/jzioyk40mhs6r2

Let's go to tophat for attendance!

https://app.tophat.com/e/245218

- **October 29:** Midterm exam
  - **The first 10 lectures will be covered in the exam!**
  - Please be here by 12:45 and start the hub
  - Report any issues to Isabel Restrepo
  - I will send out a github classroom invitation link around 12:45
  - DO NOT click on the link before 1pm
  - push your solution back to the repo and submit your pdf on gradescope by 2:20pm
  - The next lecture starts here at 2:30pm, we need to be out by 2:25pm.

- **October 31:** Guest lecturer August Guang from CCV
  - **Is this a tipping point or am I just biased?**
  - The term “tipping point” has become part of everyday English language, used to describe any kind of dramatic shift from which there is no return. However, actually defining and predicting a tipping point is extremely difficult, with the phenomenon often only becoming apparent post-hoc. Nevertheless, organizations make policy decisions based on the consequences of perceived tipping points. In light of this, we sought to understand: what characteristics make an individual more or less likely to declare a tipping point? Using logistic and random forest regression on survey data from 178 undergraduate and graduate students, we find that the perception of tipping points is primarily dependent on the characteristics of the graph, and secondarily on their own experience or emotions about tipping points. These conclusions have implications for management and sensemaking of perceived tipping points.

Mud card

- **Can you give us more info about the test? how many questions? difficultly level compared to homework? how much time are we given?**
    - 4 questions
    - difficulty level is easier than the home work
    - according to the TAs 1 hour should be enough to complete it so you have some buffer time

- **Mcar test**
    - see lecture on September 24 for paper and 26 for code
    - the mcar test takes a pandas dataframe of all numerical and np.nan values
    - the null hypothesis is that the missingness pattern is consistent with MCAR
        - if p value is large, null hypothesis is kept
        - if p value is small (p < 0.05), null hypothesis is rejected

- **missing data**
    - see lecture on September 24 for code
    - when you decide how to handle missing values, do three things first:
        - calculate the mcar p value
        - calculate what fraction of points have nans
        - calculate the fraction of nans in each feature
    - if p value is larger than 0.05 and only a few percent of points have missing values
        - drop rows or do multivariate imputation
    - if p value is small and/or a large fraction of points have missing values
        - you cannot drop rows
        - if multivariate imputation makes sense, impute
        - if it does not, leave nans as is, we will cover that in November
    - if one or a few features contain a large fraction of missing values (maybe 90% or above)
        - if it make sense to drop the columns, do so
        - if there are missing values left, repeat procedure without those features

- **how to collect preprocessed features into a new data frame**
    - if all features need to be preprocessed one way or another:

In [ ]:

```
# collect all features
cont_ftrs = []
ordinal_ftrs = []
ordinal_cats = []
cat_ftrs = []

# use the transformers and create new dataframes for each feature type
df_cont
df_ord
df_cat
# concatenate them
df_prep = pd.concat([df_cont,df_ord,df_cat])
```

- if only some features need to preprocessed:

In [ ]:

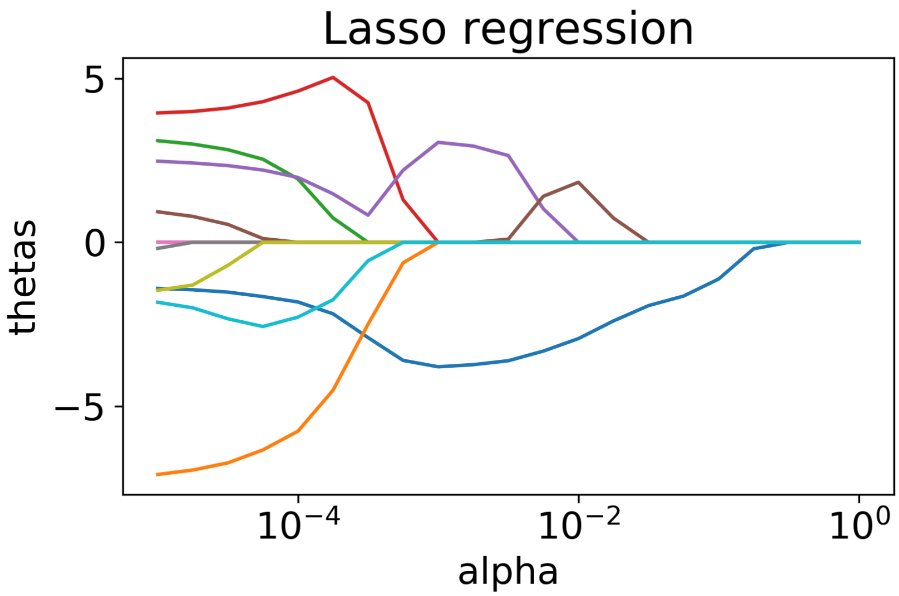
```
cat_ftrs = []
# change the features in the original df and update the dataframe
df[cat_ftrs] = scaled_feature_values # this is a numpy array
```

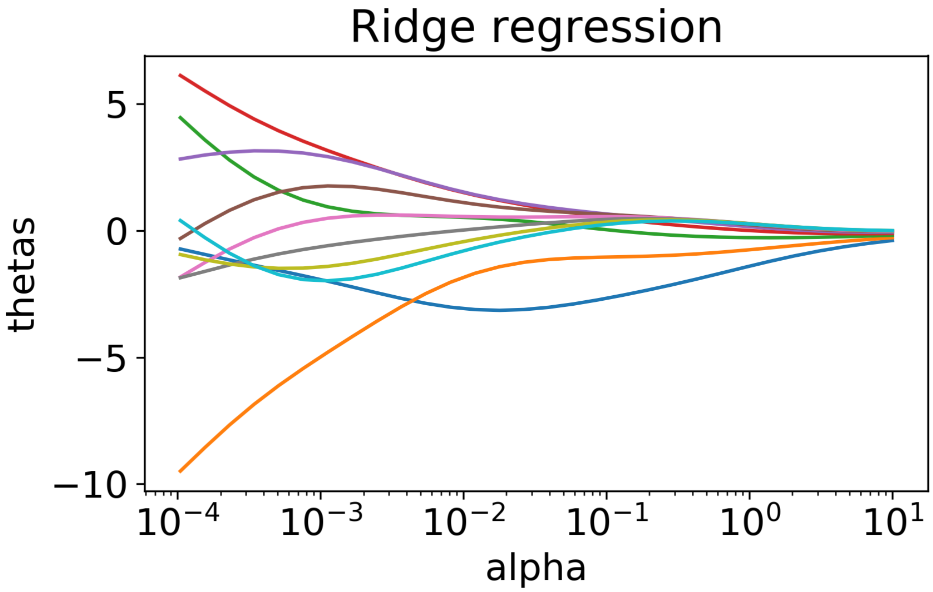
- **when to make EDA figures with the original data versus the preprocessed data**
    - tough question, it is case-specific
    - continuous features: usually show original data
    - categorical and ordinal features, it might make sense to show preprocessed data

- **What's the difference between ridge and lasso regression?**
    - both are linear regression methods with regularization
        - regularization can help you to avoid overfitting
    - lasso (https://scikit-learn.org/stable/modules/linear_model.html#lasso): regularization is done with the l1 norm of theta ($\frac{\alpha}{m} \sum_{j=0}^{m} |\theta_j|$)
        - good for feature selection as well because if alpha is large, some thetas are 0
    - ridge (https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression): regularization is done with the l2 norm of theta ($\frac{\alpha}{m} \sum_{j=0}^{m} \theta_j^2$)
        - l2 regularization produces smoothly varying thetas as a function of alpha

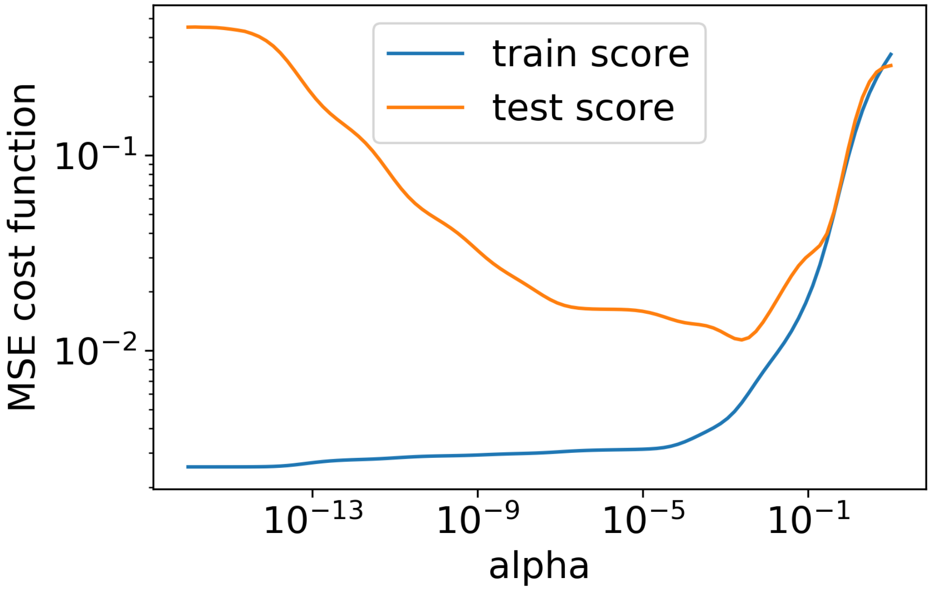
- **Are there other cost function for logistic regression other than the log loss function?**
    - not in sklearn but in principle you could construct other cost functions
    - cost function inputs:
        - the true values of the target variable
        - the feature matrix
        - model parameters
    - cost function output:
        - a single number that measures how well the model performs on the data given the model parameters

- **In the first plot where we looked at the lasso regression, why did one of the features go to zero and then later on show up back again? how should this be interpreted?**
  - that can happen
  - it's OK because only one alpha value optimizies performance on the test set and you really only care about the corresponding theta coefficients
  - in l2, some coefficients turn from negative to positive and vice versa.





- **Can you go over bias and variance again, specifically what they mean in the context of the success of a model, and how they relate to metrics that we can look at, such as mse.**
  - step 1: split your data to train and test
  - step 2: choose an evaluation metric
  - step 3: choose a model hyperparameter (e.g., the regularization parameter)
  - step 4: loop through a wide range of hyperparameter values
  - step 5: plot the tran and test scores
  - high bias: model with a hyperparameter value performs poorly on both train and test
  - high variace: model with a hyperparameter value perform very good on train but poorly on test
    - model doesn't generalize to new points



- **In logistic regression, what is penalty='l1' and what is c=1/alpha**
  - 'l1' is regularization using the l1 norm of theta
  - 'l2' is regularization using the l2 norm of theta
  - C is the inverse of the regularization parameter, alpha

- **Can logistic regression be used for continuous target variables?**
  - No. Logistic regression is a classification method.
  - sklearn's logistic regression expects classification labels as the target variable (values between 0 and n_classes -1)

HW time!