Outline slide for Wed Feb 1

- Note: (using the same note pack for Mon/Wed in W5!)
- **Brief review: errors, concept of bias and variance in modeling**
- **Review decision boundaries**
- **Discuss minimum error in classification problems (Bayes boundary)**
- **Discuss logistic regression as a classification method**
- **Project discussions**

Data Science Methods for Clean Energy Research

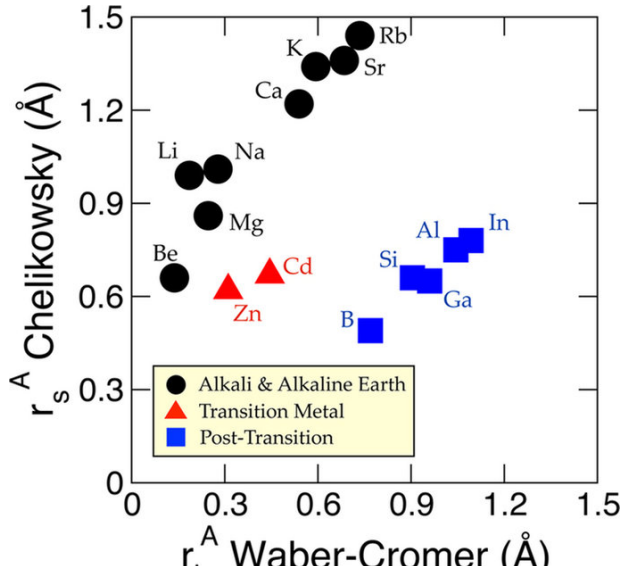
Week 5 , Lecture 1-1.5: Machine Learning
Introduction and Classification

January 24, 2017

UNIVERSITY of WASHINGTON

Motivation

- **You have some computational assessment of various elements. Using a combination of the data, you can see a clear relationship** classification **of the elemental category**
- **How would you use this information to make** predictions **about the classification of other elements?**



Outline

- Reminder about reading (ISL text) + projects!
- Warmup
- Overview of key terms/vocab in ML
  - Types of learning / modeling
  - Different models, different purpose(s)
  - Types of data
  - Evaluating your models
- Classification methods
  - KNN + Python practice
  - Logistic regression

Figures and notation will follow Introduction to Statistical Learning unless otherwise noted; see #dsmcer for more info or Google for free pdf of book

Warm up

- Write down one question you have about machine learning (2-min)
- Discuss at your table and decide if there is one question you'd like the answer to (2-min)
  - Send DM to me on Slack
- JP or DACB will answer the questions (<= 5min)

Concepts: Types of learning and purpose of learning (basic terminology)

- **Differentiating** “statistical learning” **vs** “machine learning”
- **Differentiating** “supervised” **vs** “unsupervised”
- **Differentiating** “prediction” **vs** “inference”

Some concepts about supervised learning (basic terminology)

- In supervised learning we assume there is a functional form that can describe our output vector (Y) given some input vector (X)

$$Y = f(X) + \epsilon. \qquad (2.1)$$

- We are concerned with estimating a function that can predict an output given a certain set of inputs

$$\hat{Y} = \hat{f}(X), \qquad (2.2)$$

- There are two limits to the error in our function

$$\begin{aligned} E(Y - \hat{Y})^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= \underbrace{[f(X) - \hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{Irreducible}}, \end{aligned} \qquad (2.3)$$

"E()" is the expectation value or average squared error

Some concepts about supervised learning (functional form)

- **Some models have a defined functional form over the entire range of input values** (parametric modeling)
  - **We assume a functional form** (parameters or coefficients)
  - **We determine the parameters by minimizing the error between the model and reality for some subset of our data** (fitting or training)

$$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p. \qquad (2.4)$$

- **Some models do not assume a defined functional form – the goal is just to get as close as possible to the data** (non-parametric modeling)
  - **There is still a model fitting procedure; the model still has some information content that is determined from your data**

- **In parametric modeling you risk** overfitting **especially with** noisy data**. In non-parametric models you require much larger training data sets**

Some concepts about supervised learning (I/O type)

- **In supervised learning, some models deal with the prediction of a numerical output given some inputs:** regression problems
  - **See if you can come up w/an example of where regression would be useful in materials / energy science**
- **On the other hand, some models deal with the prediction of whether some object (defined by** X**) is a member of a certain group (**or class**):** classification problems
  - **See if you can come up w/an example of where classification would be useful in a problem related to social science**
  - **Some methods (**e.g., KNN**) can be used for regression purposes and classification purposes! Weird!**

Basic Concepts about flexibility and interpretability

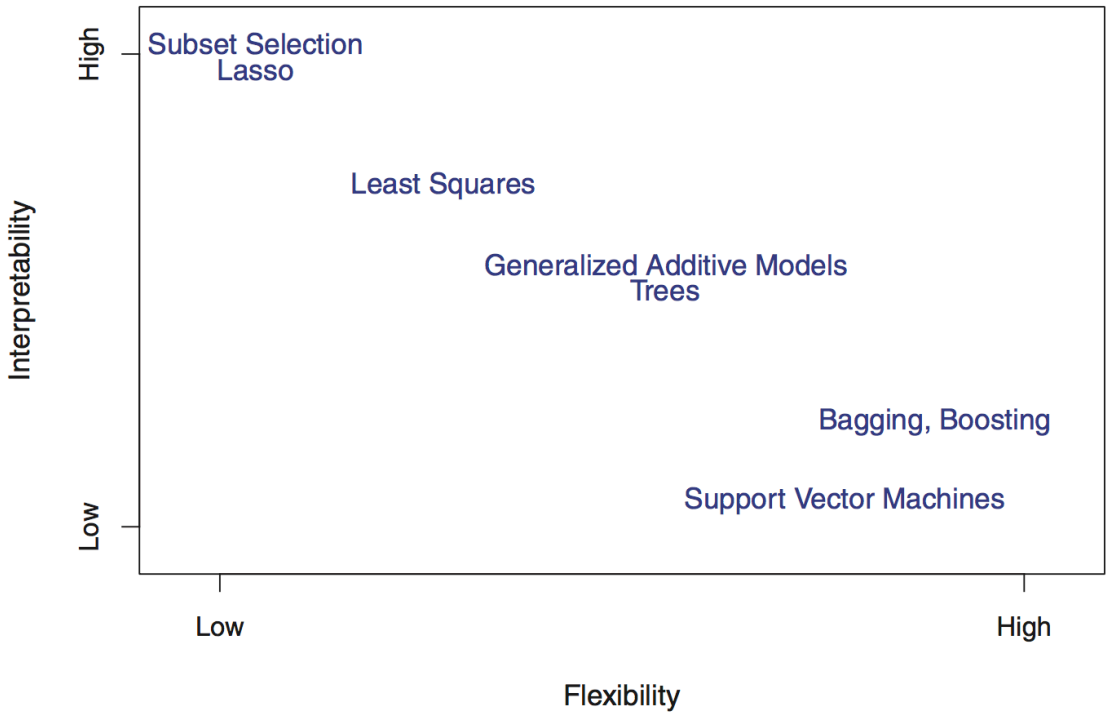


FIGURE 2.7. *A representation of the tradeoff between flexibility and interpretability, using different statistical learning methods. In general, as the flexibility of a method increases, its interpretability decreases.*

Data: Types of data and purpose

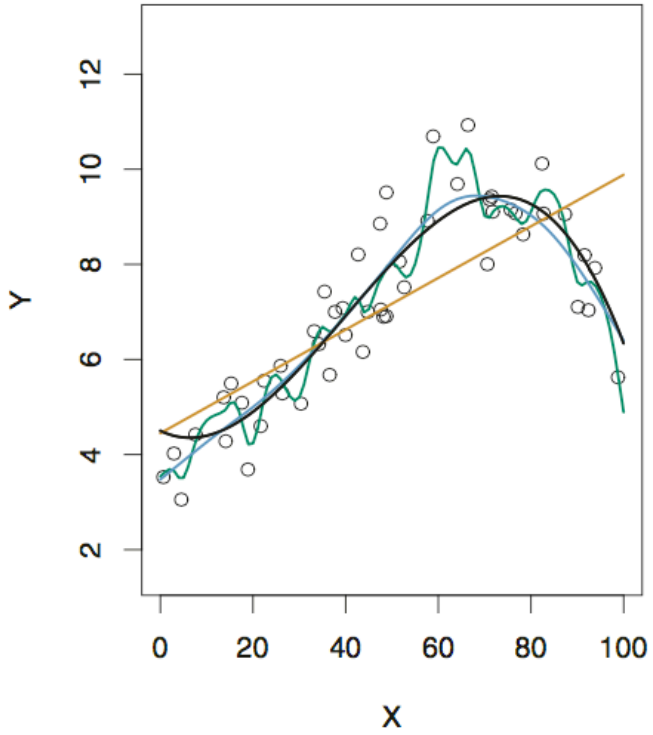
- **Supervised learning problems** always **require a set of data that must be used to** train **your model**
  - **Remember: model training means we are finding the minimum error between our predicted outputs and measured outputs (**$\hat{Y}$ vs Y**)**
- **We always want to know how our model is doing "out in the real world" on some data that the training process was blind to (**validation**)**
- **Do unsupervised learning models need training and validation sets?**

Assessment: Measuring model accuracy (regression)

- The MSE as an estimator $$E\left(y_0 - \hat{f}(x_0)\right)^2$$
- Challenges with the MSE and why we need other estimators
  - Side conversation: Why is it squared error?
- Does anyone know why we can't use the MSE in classification problems?

Assessment: Comparing models in linear regression

- **The true functional form is given w/the** black **curve**
- **Compare errors in the** training **data from three test functions (**orange, green, blue**) [**grey curve, right**]**
- **Compare errors in the** testing **data from the three test functions [**red curve, right**]**



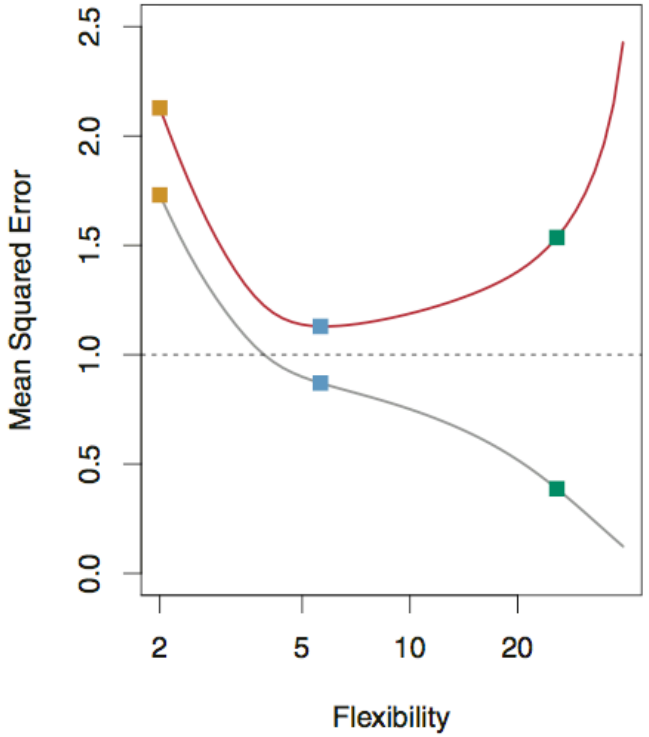


Fig 2.9, ISL

Assessment: Bias and variance

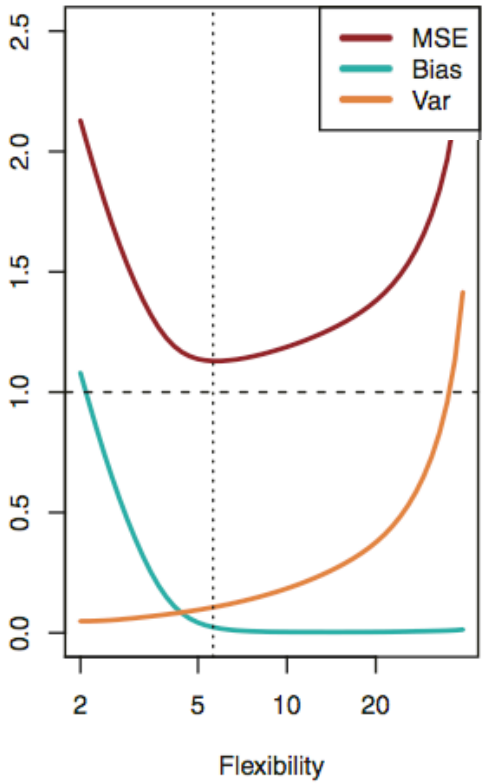
- **Ultimately we care about the true expectation value of the error of our model (given infinite testing data)**

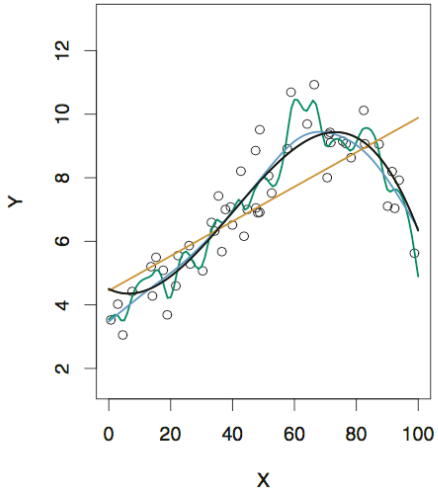
$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon). \qquad (2.7)$$

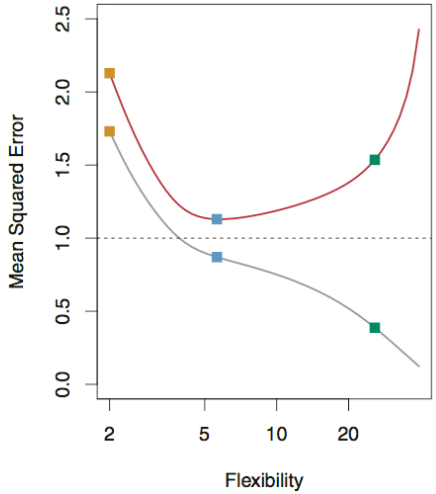
- **The** variance **in the error shows us how much the error changes if we estimated f with a different set of** training data **(e.g., consider fit w/many splines)**
- **The** bias **in the error shows us how much error is introduced by the simplicity of our model (e.g., curvy data w/linear fit)**
- **The total MSE can never fall below the variance in our observations! (**irreducible error**)**

Bias and variance: the tradeoff

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon). \qquad (2.7)$$







In practice, the bias error is difficult to estimate quantitatively, but the principle is critical to qualitatively understand!

Modified Fig 2.12, ISL

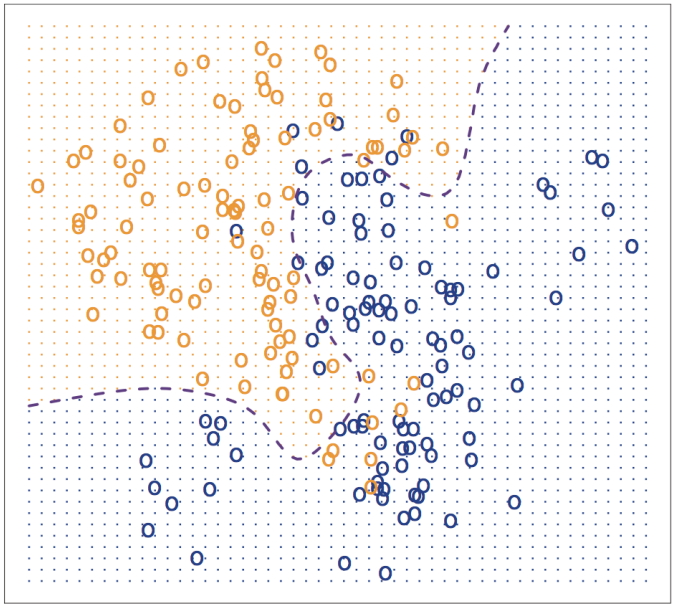
See figs 2.9-2.11 and full version of Fig 2.12 in ISL for more examples!

Assessment: Errors in classification problems

- **Even though we aren't making a quantitative prediction, we still need to assess the model accuracy**
- **Consider the frequency of** incorrect classification**, the error rate**
  - **The** training error rate **is the frequency of errors on our training data**
  - **Generally more interested in the** test error rate**, errors on data that were not used to train our classifier**

An estimator for the irreducible error in classification: Bayes classifier

- **My first classifier!**
- **In principle if we know the true** conditional probability distribution **for our data, we can construct a surface (**a decision boundary**) that guarantees our error rate in classification is minimized**
- **This ideal model is the "**Bayes classifier**" and this boundary is the "**Bayes decision boundary**"**



Brief discussion of decision boundaries, visualization of classification models, and a warning about dimensionality!

$$\Pr(Y = k|X = x)$$

FIGURE 2.13. *A simulated data set consisting of* 100 *observations in each of two groups, indicated in blue and in orange. The purple dashed line represents the Bayes decision boundary. The orange background grid indicates the region in which a test observation will be assigned to the orange class, and the blue background grid indicates the region in which a test observation will be assigned to the blue class.*

Transition from ML overview!

- Any questions / concerns
- If people are really bored, someone could ask me to tell a joke right now

Classification methods

- [quick warmup?] **Examples of classification problems in materials or molecular science?**
- **We will cover two types of classification methods in this class:**
  - **k Nearest Neighbors (often KNN)**
    - **Making a map of parameter space and estimating the class membership based on locality to other members of the class**
  - **Logistic regression**
    - **Making a mathematic guess about the likelihood of class membership using a simplified mathematical model (the logistic function)**

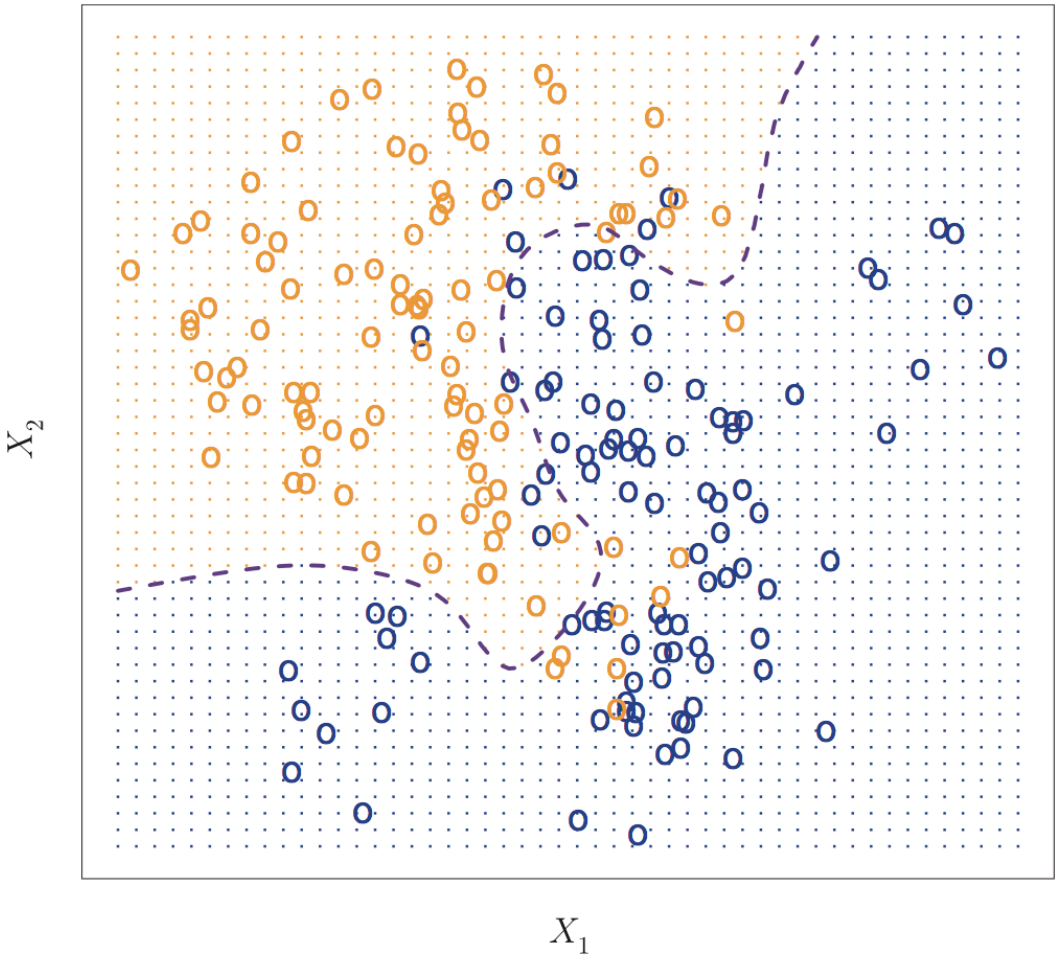
$$\Pr(Y = k|X = x)$$

(probably) the simplest classification method is k-nearest neighbors (KNN)

- The KNN classifier works on assigning the probability of membership in a class based on distance to other members in a class
  - Simple, right?
  - Our selection is based only on K – the number of neighbors we consider
  - What are the possible ranges of K?
  - The KNN classifier

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j). \qquad (2.12)$$

A note about distances and identifying who is "nearest"?



- In this example, and all others we looked at and discussed you all made a key assumption (*without realizing it*) when you were identifying neighbors
- The data in each dimension of X should be scaled similarly!
- We are concerned with relative distances in the parameter space, not absolute distances in the underlying units
- Otherwise units, dimensionality, etc. can drastically favor closeness in arbitrary dimensions of X

Illustration of KNN

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j). \qquad (2.12)$$

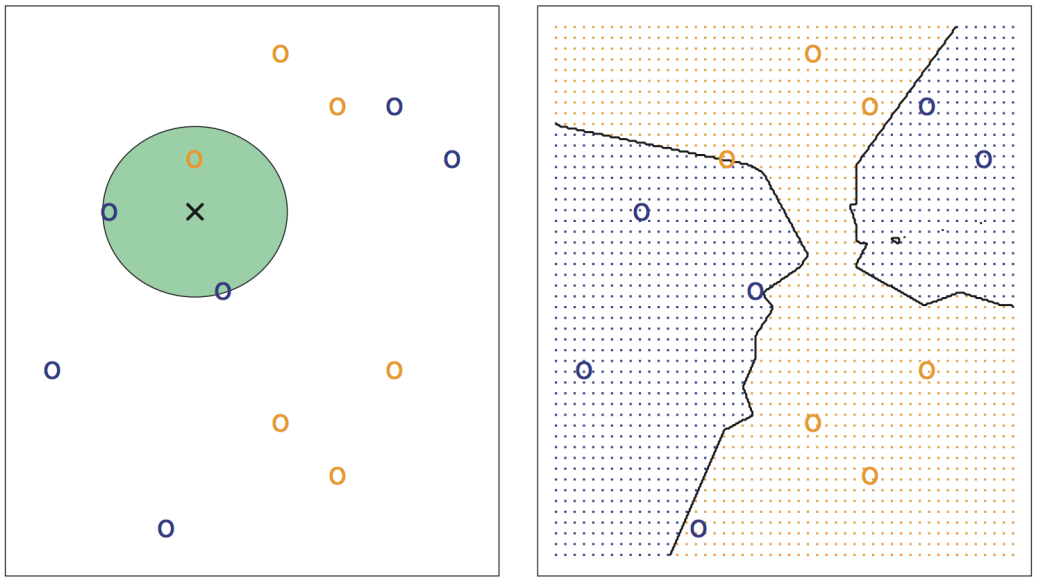


FIGURE 2.14. *The KNN approach, using $K = 3$, is illustrated in a simple situation with six blue observations and six orange observations.* Left: *a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue.* Right: *The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.*

Effect of K on boundary

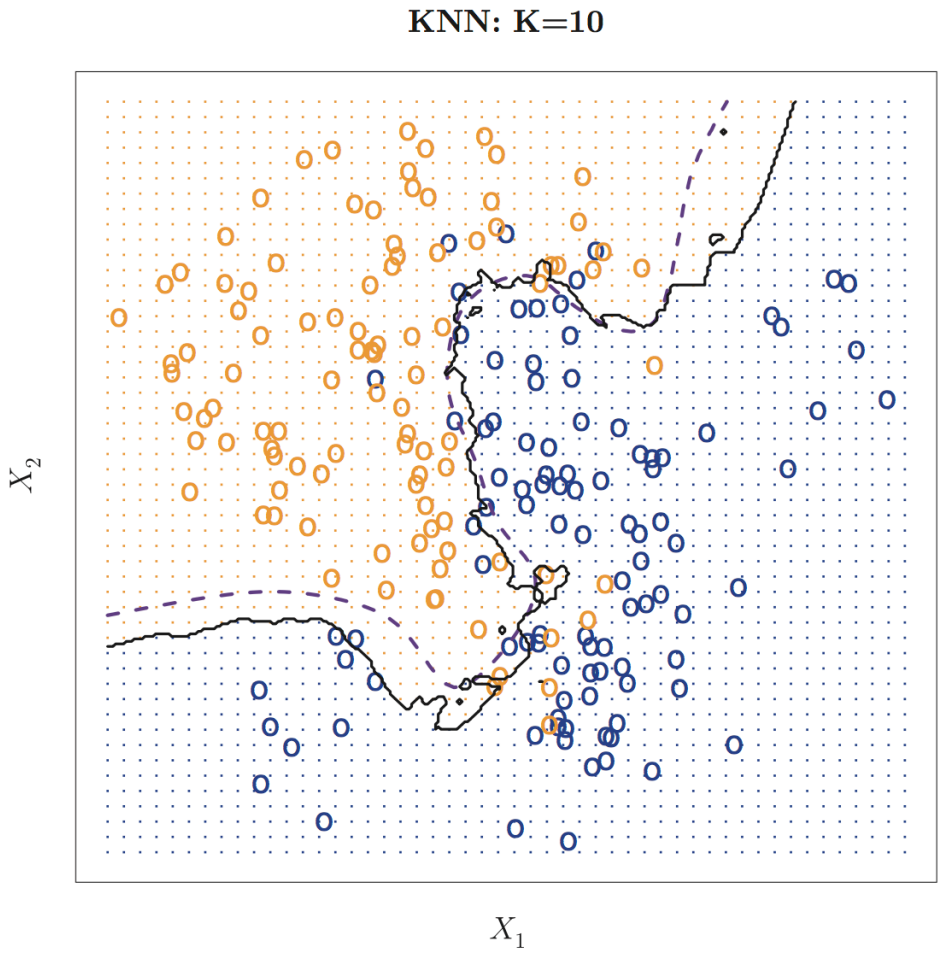


FIGURE 2.15. *The black curve indicates the KNN decision boundary on the data from Figure 2.13, using $K = 10$. The Bayes decision boundary is shown as a purple dashed line. The KNN and Bayes decision boundaries are very similar.*

K too big or too small: noise and accuracy

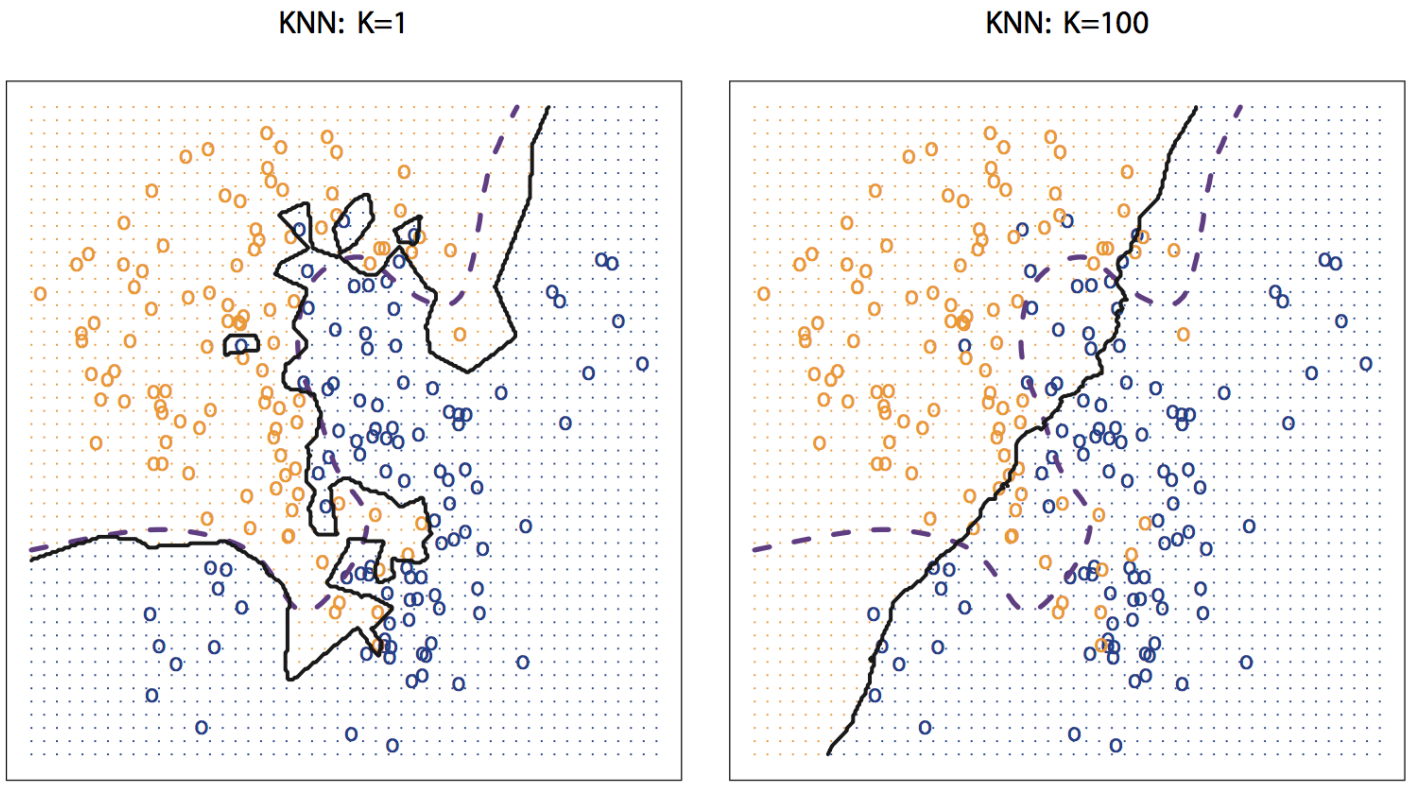


FIGURE 2.16. *A comparison of the KNN decision boundaries (solid black curves) obtained using $K = 1$ and $K = 100$ on the data from Figure 2.13. With $K = 1$, the decision boundary is overly flexible, while with $K = 100$ it is not sufficiently flexible. The Bayes decision boundary is shown as a purple dashed line.*

Effect of K on accuracy

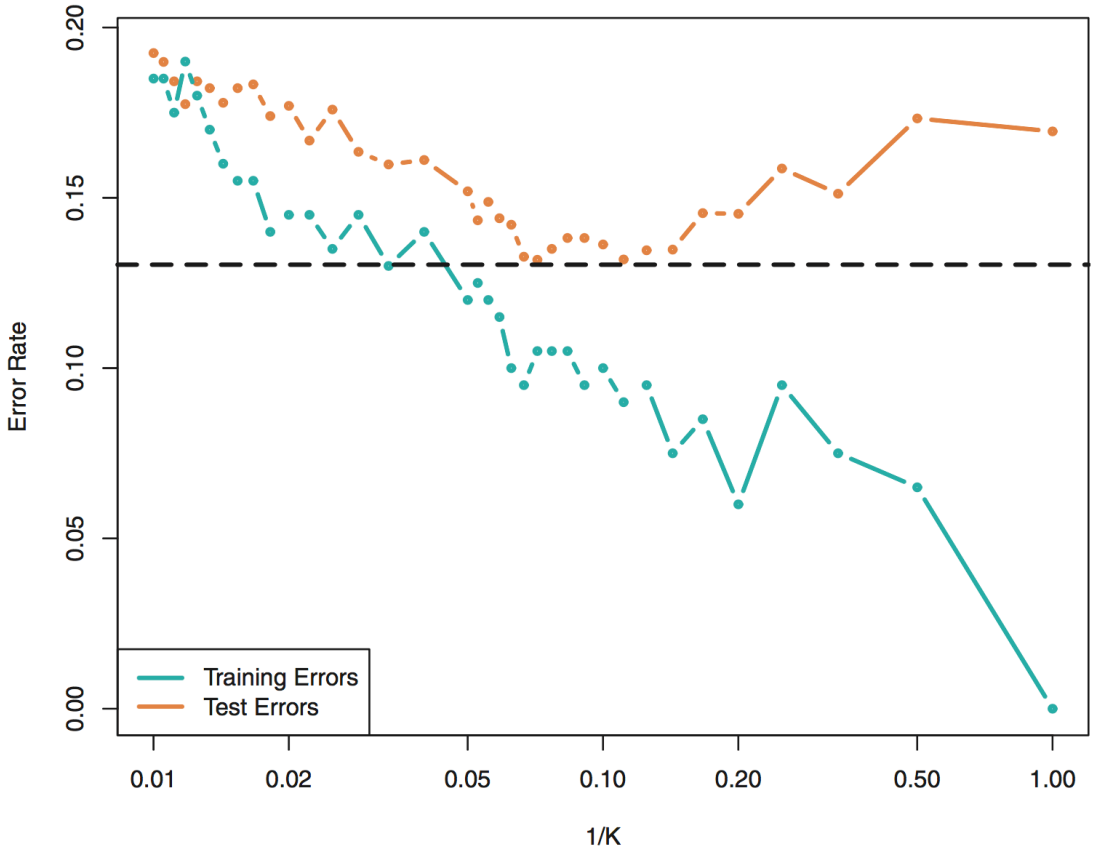


FIGURE 2.17. *The KNN training error rate (blue, 200 observations) and test error rate (orange, 5,000 observations) on the data from Figure 2.13, as the level of flexibility (assessed using $1/K$) increases, or equivalently as the number of neighbors K decreases. The black dashed line indicates the Bayes error rate. The jumpiness of the curves is due to the small size of the training data set.*

Practice with the KNN!

- Lets get our hands dirty
- The example data we are using come from the paper below
- Much of the code comes from the scikit-learn KNN tutorial (http://scikit-learn.org/stable/tutorial/statistical_inference/supervised_learning.html)

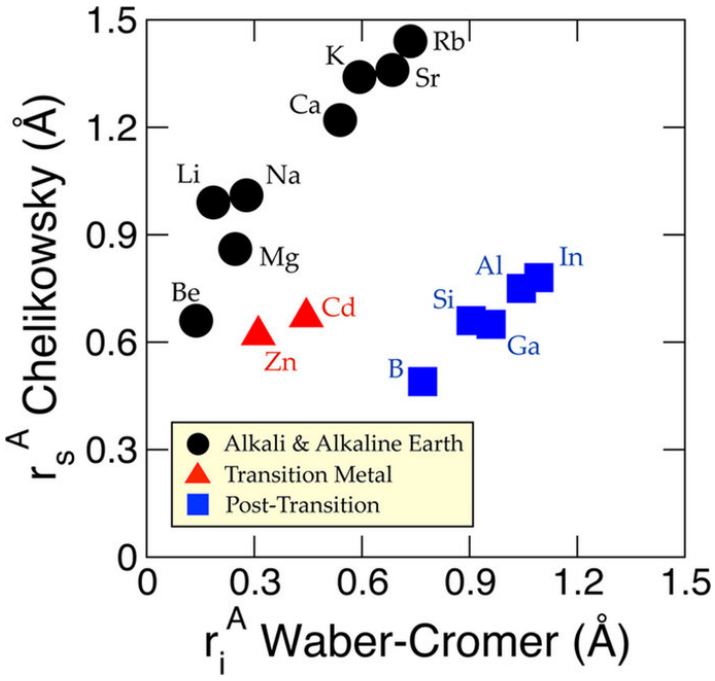
Materials Prediction via Classification Learning

Prasanna V. Balachandran, James Theiler, James M. Rondinelli & Turab Lookman

Scientific Reports **5**, Article number: 13285 (2015)
doi:10.1038/srep13285
Download Citation

Electronic properties and materials
Mechanical properties

Received: 02 April 2015
Accepted: 16 July 2015
Published online: 25 August 2015



With one partner (14x2!)

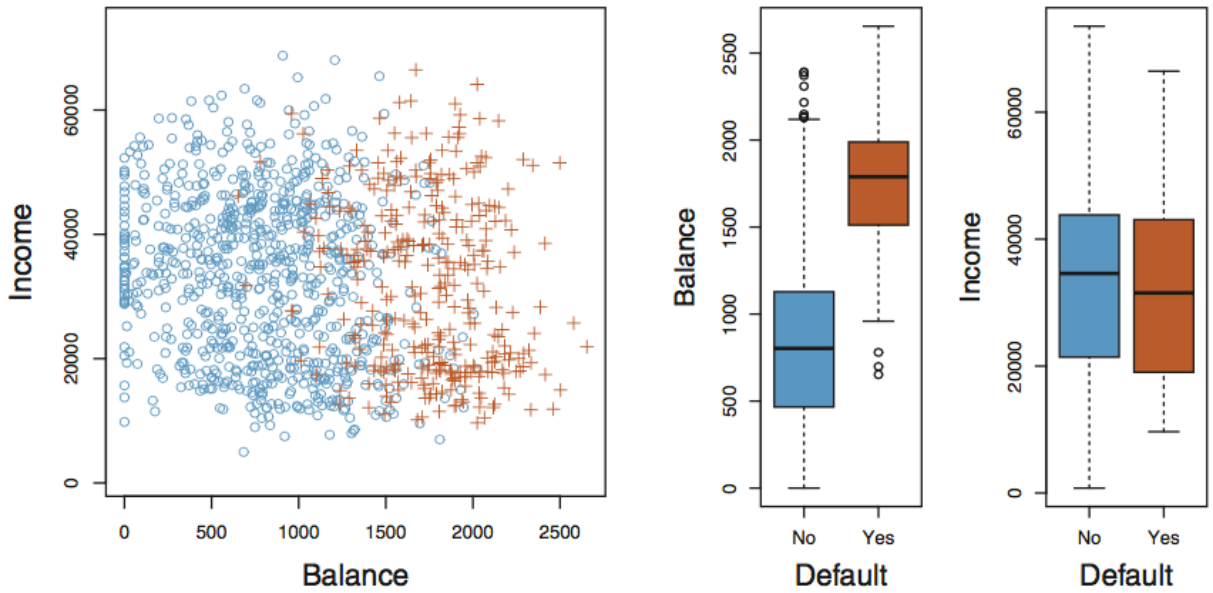
- Get the W5L1.zip from #dsmcer and load it on one computer – it contains two data files and a python notebook
- Open the notebook
- Follow the instructions
- OK?

Classification with the logistic model

Key concepts

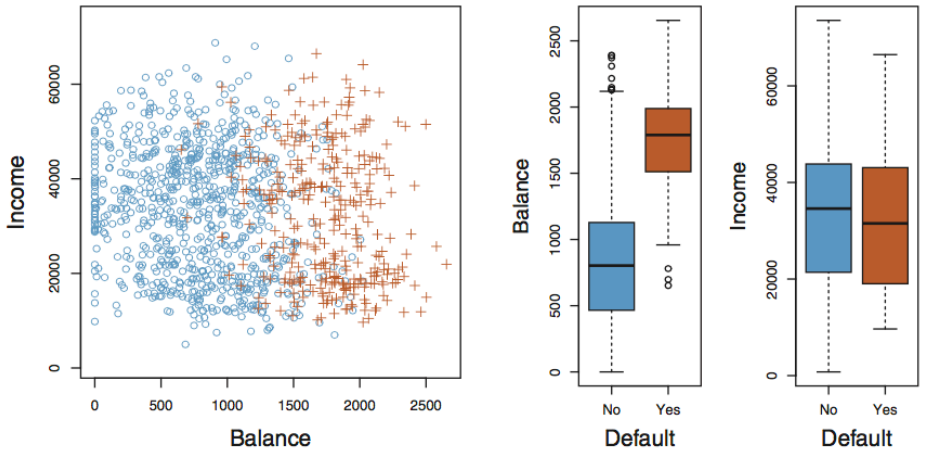
Consider the data below (ISL Fig 4.1), we are trying to make a guess as to whether someone will default on a loan on the basis of their bank account balance and income level

Questions?



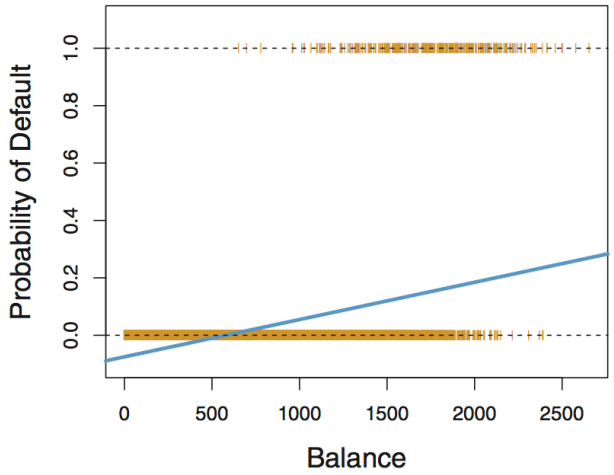
Classification with the logistic model

- **As we said in class (Mon 2/1), performing regression on discrete classes is arbitrary (**what are the outputs we assign to Yes/No and why?**)**
- **One caveat is the binary (two level response). Here we can always use a 0/1 coding!**
- **Lets look at simple regression on this data**



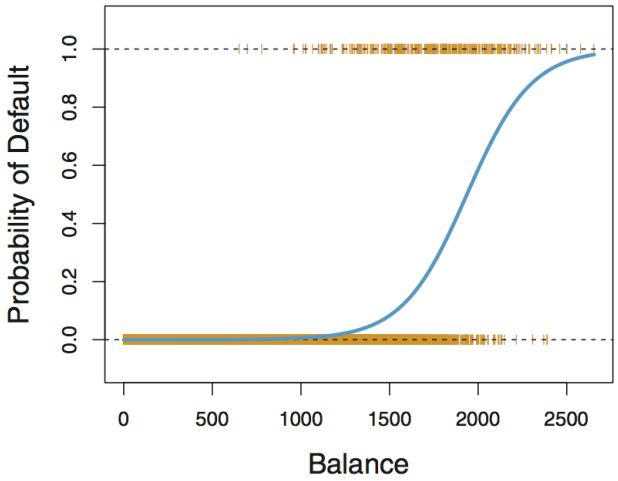
Classification with the logistic model

- **Fig 4.2 (left) shows us the outcome of simple linear regression using a 0/1 coding for the probability of default (0.0 or 1.0)**
- **What does the model predict for Balance < $500?**



- **Fig 4.2 (right) shows us the outcome of regression using a different functional form to describe situation where our variables are bounded between 0 and 1**
- **This is modeled using the** logistic function**, which lets us sensibly answer this question: "**what is the probability of default given some balance?**"**

$$\Pr(\texttt{default} = \texttt{Yes} | \texttt{balance})$$



The logistic function

- **The logistic equation (4.2) provides a smooth transition between 0 and 1**

$$p(X) = \beta_0 + \beta_1 X. \qquad (4.1)$$

Compare simple linear model and logistic eq

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}. \qquad (4.2)$$

$$\frac{p(X)}{1 - p(X)} = e^{\beta_0+\beta_1 X}. \qquad (4.3)$$

In terms of the "odds"

Can anyone define "odds"? If your probability of success is 1 in 5 (0.2) , your odds of success are 1 in 4: 0.2 / (1-.2) = .25

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X. \qquad (4.4)$$

Log of the odds is the "logit", relates the change in log of odds of success w/change in X

Is the logistic model a classifier?

- **Yes!**
- **Given a class situation "e.g., default vs. non-default", the logistic model can take a set of** training data **and give a function that makes a** prediction **about what class a new or different input would be in**
  - $P < 0.5$ = no default **vs.** $P >= 0.5$ default
- **We can also use a binary response (e.g., 0 vs 1) and train a** multiple logistic **model (e.g., many** X_i**)**
- **We could also use** multinomial logistic **models to describe a situation with >2 classes**

Basic concept of logistic regression (much more next two weeks on regression!)

- **For reasons we won't discuss in detail standard regression models don't work well on the logistic equation.**
- **But we still need to determine our coefficients: β_i**
- **This is done through a procedure called** maximum likelihood estimation
  - **The basic idea is that we seek to find the set of β_i that maximizes the probability of making the correct prediction**

Simple illustration of MLE

- The likelihood function can be defined in terms of the product of all the probabilities
  - In the case of the binary (0 vs 1) scenario:

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})). \tag{4.5}$$

  - Recall each probability is defined by our logistic eq:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}. \tag{4.2}$$

  - We seek the values of β_i that maximize 4.5

Example and implementation in Python

- We saw on Monday that there is a python library "sklearn" (sci-kit learn), that can be used to perform machine learning techniques
  - Monday example: KNN
  - We will use many other ML methods from sklearn this quarter
  - Sometimes practice in class, sometimes practice on HW, sometimes example in class notes

Example and implementation in Python

> Recall key elements of our KNN model using sklearn:

```
In [ ]: # load library, create an object for the learning algorithm, run the fit, make a prediction
        from sklearn.neighbors import KNeighborsClassifier
        knn = KNeighborsClassifier(n_neighbors=5)
        knn.fit(elements[['rWC','rCh']],np.ravel(elements.Type))
        knn.predict(Xtest)
```

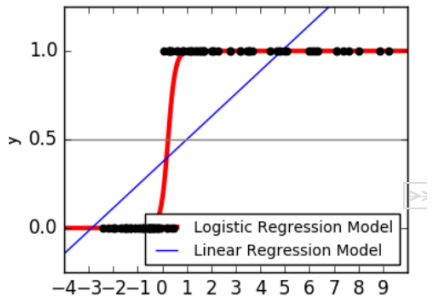
> Logistic regression is similar:

- From sklearn tutorial

For classification, as in the labeling iris task, linear regression is not the right approach as it will give too much weight to data far from the decision frontier. A linear approach is to fit a sigmoid function or **logistic** function:

$$y = \text{sigmoid}(X\beta - \text{offset}) + \epsilon = \frac{1}{1 + \exp(-X\beta + \text{offset})} + \epsilon$$

```
logistic = linear_model.LogisticRegression(C=1e5)
logistic.fit(iris_X_train, iris_y_train)
```



SEDS/DSCMER project work (day 1)

UNIVERSITY of WASHINGTON

Project overview

- **Project description and requirement: Your DIRECT course project will be performed in teams of 4 (there may be one team of 3) with a goal of giving you practical experience in software development and application of Data Science tools to a problem of practical interest.**

- **The projects are nearly completely open-ended! You could work with some of the data sets we have talked about it class, find your own data, use data from research. What you do with the data is up to you as well! Outcome: A team based piece of software (written in Python and developed on GitHub) that uses SEDS and DSMCER course materials on a project related to materials and/or clean energy. Projects should include a "complex" data set (definition of "complexity" subject to instructor approval)**

- **You will work on project teams that will be assigned according to the 2/1,2/6,2/8 class sessions. The projects should have significant scope and take several weeks to complete (which is why we are starting them now!)**

Project milestones

- Wed 2/1: Project concept and individual research
- Mon 2/6: Project pitches and forming teams
- Wed 2/8: Team pitches of basic idea
- SEDS HW4
- SEDS HW5
- Poster session: Presentation w/DIRECT faculty, (maybe) project sponsors, your labmates and your classmates. Location and time: TBD
- Final commit on GitHub: Turning in the project. Due Wed March 15 at 5pm

DRAFT Project rubric
(open to comments!)

Item	SEDS Points	DSMCER Points	Scheme
Project scoping phase (2/1-2/8 classes): Participation in project scoping and team-forming process and presentations	5	5	Individual
Quality and effectiveness of final code (assessed by Dave Beck): Does the code do what it set out to do, were SEDS principles followed, is it developed in a sensible manner that allows for future developments and contributions	40	0	Group
Effectiveness of the project in addressing a research or engineering challenge related to materials, clean energy, or both	0	20	Group
Effectiveness of project in utilizing one or more Data Science methods (statistical/machine learning or advanced visualization) in addressing a challenge related to materials, clean energy, or both	0	20	Group
Quality of final presentation (poster and 5-min pitch/overview)	10	10	Group
Contributions to project as evidenced by GitHub activity	10	10	Individual
Peer evaluation	15	15	Individual
Overall assessment of originality: We don't want you to repeat an existing tool (e.g., don’t' make your own KNN code when it exists in scikit-learn!)	5	5	Group
Interesting, novel or original use of data	5	5	Group
Individual instructor grades	10	10	Individual
Total	100	100	

Plan for next 3 DSMCER sessions

- **Wed 2/1:** 40 min of class time (2:10-2:50). Explain projects, timetable, etc. Answer any questions, independent time researching projects and discussing ideas on Slack. May pre-form groups of 2 if interested (no larger).

- **Monday 2/6:** 40 min of class time (2:10-2:50). Opportunity for enterprising students to "pitch" ideas and recruit team members. We especially ESL students to participate and note we recognize their skills / abilities even if they are still new at English – great opportunity to practice and get feedback! Students will contribute to a Google Sheet if they like a project and want to sign up. Dave and I will form project teams @ end of class

- **Wed 2/8:** 40 min of class time (2:10-2:50). Teams have to pitch ideas to class/Dave/Jim for feedback. Two slide overview of project using Google slides template (so we can just have one laptop). We will provide a template

A quick lit review of an interesting ML paper

Statistics and Machine Learning

A recent example of ML applied to materials property prediction

> **Highlights from Abstract (10,000 foot view)**
  - Kernel regression **is used as a rigorous and** nonparametric **statistical technique to** predict properties **of atomic crystals**
  - **The property prediction approach is illustrated by** training models **to predict** electronic **properties of 746 binary metal oxides and** elastic **properties of 1173 crystals**
  - **As a first approach to solving the** inverse problem**, an exhaustive** enumeration algorithm **is described**

From Wikipedia: Non-parametric models differ from parametric models in that the model structure is not specified a priori but is instead determined from data. The term non-parametric is not meant to imply that such models completely lack parameters but that the number and nature of the parameters are flexible and not fixed in advance.

Result/Concept: Parsing out your data

- **This paper drew data from the Materials Project using their API.**
  - The data are calculated using DFT [any questions/concerns?]
- **In a regression or ML model data need to be divided into training and validation sets**
  - **Basic principles: if you don't save some of your data how can you ever** quantitatively evaluate **whether your model is predictive?**
    - **Sometimes training is broken into an additional piece (testing → more later)**
    - **Sometimes multiple rounds of training and validation are used (more later)**

Result/Concept: Types of methods and assessment

- **This paper uses a machine learning model called kernel ridge regression. Much more later. The basic idea is a regression technique**
- **The predictors that are used to fit the model include:**
  - **The 3D crystallographic space group of the material**
    - **Example: "*P21/c*, centrosymmetric" How is this coded into a variable?**
  - **Number of atoms of each element in chemical formula**

Result/Concept: Validation (electronic properties)

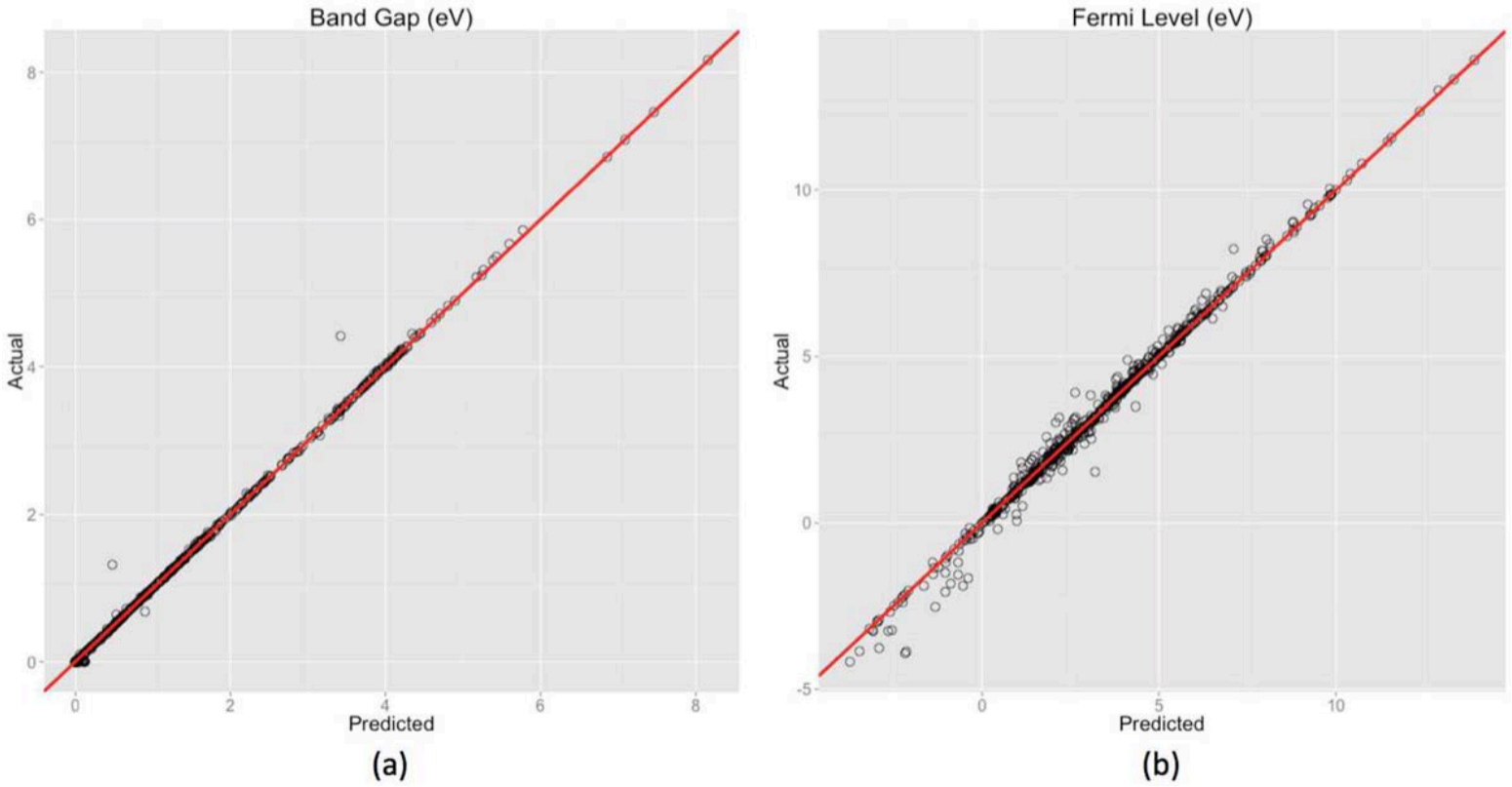


Figure 1. Actual response data vs. predicted values of (a) band gap and (b) Fermi level.

Result/Concept: Validation (elastic properties)

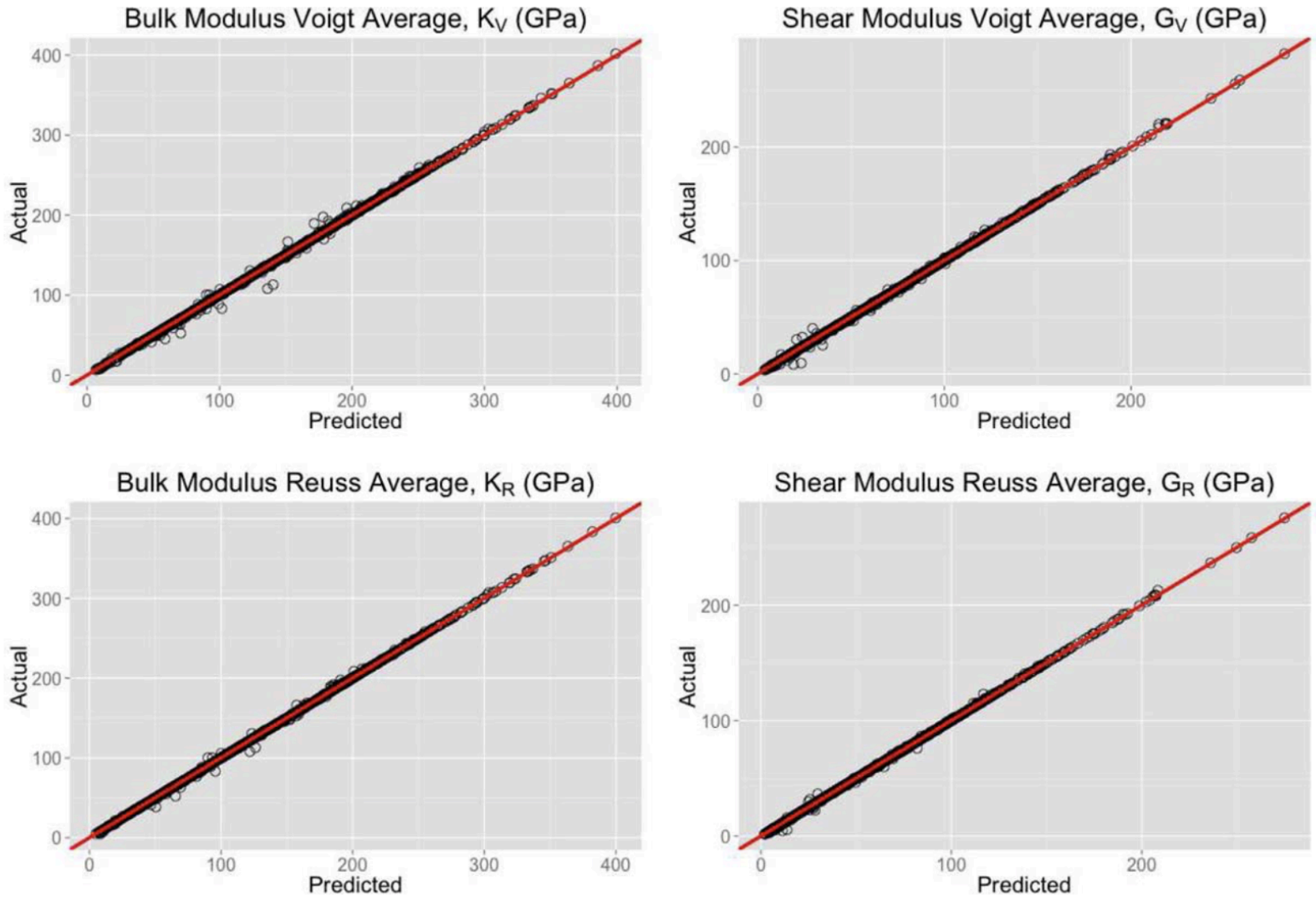


Figure 2. Actual response data vs. predicted values of bulk and shear moduli Voigt average (top), and bulk and shear moduli Reuss average (bottom).

Result/Concept: Application

- **The ML algorithm the authors chose permits a process called "exhaustive enumeration" (*just like it sounds*) on a reasonable time scale**
- **The authors used this property to carry out something called *inverse design***
  - **Why does this matter in materials / chemicals science?**

Table 4. Results (Crystal Composition and Space Group Number) of the Exhaustive Enumeration Algorithm for some Given Specifications

Specifications	Crystals
$\lvert\tilde{Y}_{\text{band\_gap}} - 1.23\rvert \leq 10^{-2}$	Bi_2O_3 (#197), Mn_7O_{25} (#164)*
$\min Y_{\text{band\_gap}}$	Rb_9O_{13} (#167)*
$0.5\,\text{eV} \leq Y_{\text{efermi}} \leq 1.5\,\text{eV}$	
$\tilde{Y}_{\text{band\_gap}} \leq 2\,\text{eV}$	$Co_{21}O_{22}$ (#141)*, Tb_8O_{21} (#130)*,
$\lvert\tilde{Y}_{\text{efermi}} - 1.5\,\text{eV}\rvert \leq 10^{-2}$	CsO_{48} (#185)*

Result/Concept: Reproducibility

Algorithm 1: Pseudocode of the exhaustive enumeration algorithm to evaluate the properties of all *unique* combinations of metal oxides.

Input : List with names of properties and matrices of predictor data $X_{i,p}$ and unique combinations $\bar{X}_{i,p}$
Output: List of predictions for all unique combinations for each property

```
for i from 1 to n                                   /* Space group number loop */
do
    if X̄_{i,sg} = 1 then
        for i′ from 1 to n                                    /* Metal loop */
        do
            foreach p ∈ P \ {sg, O} do
                if X̄_{i′,sg} = 1 and X_{i′,p} ≠ 0 then
                    for i″ from 1 to n                        /* Oxygen loop */
                    do
                        if X̄_{i″,O} = 1 then
                            /* Current data point indices:                  */
                            /* Space group number i, Metal i′, Oxygen i″    */
                            Predict the properties j ∈ J of the current metal oxide;
                            Store predictions in the output list;
                        end
                    end
                end
            end
        end
    end
end
```

Also see the outstanding SI of this paper