# Statistical Learning

Sravani Vadlamani

# Agenda

- Input and Output Variables
- How do we estimate f?
- Supervised Learning vs. Unsupervised Learning
- Regression vs Classification Problems
- Prediction Accuracy vs Model Interpretability
- Bias-Variance Tradeoff

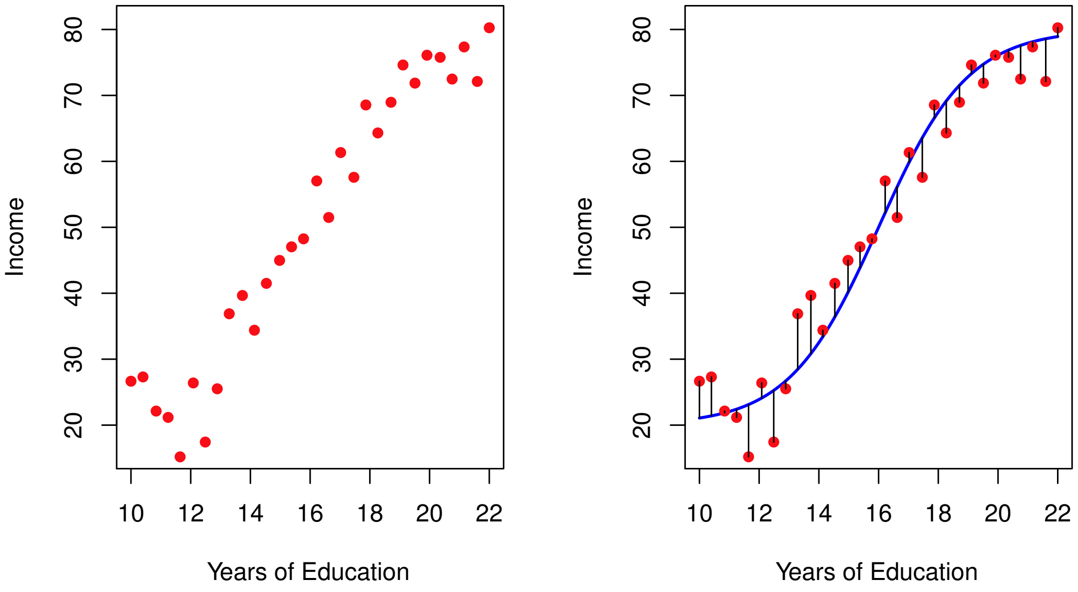
# Input and Output Variables

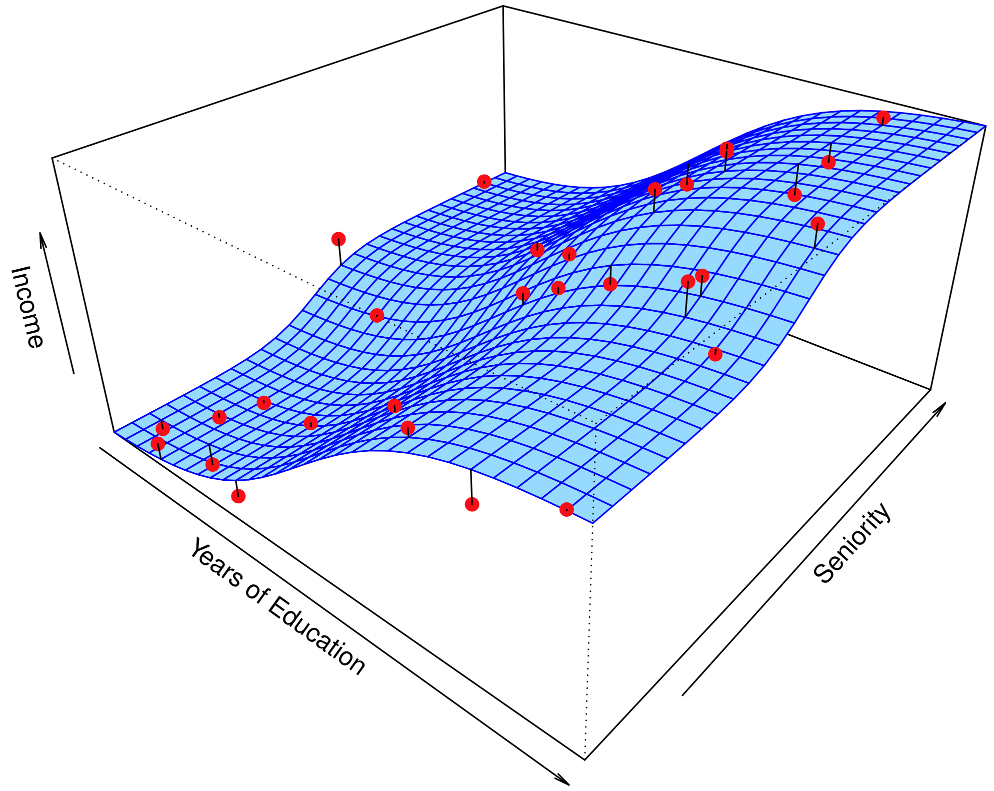
- Input variables are independent variables, predictors, features. They are denoted as $X_i$ = $(X_1 , X_2 , .... X_p )$
- Output variables are also known as response or dependent variable and is denoted as $Y_i$
- We believe there is a relationship between Y and at least one of the X's. We model the relation as

$$Y_i = f(X_i) + \varepsilon$$

- Where ***f*** is an unknown function and $\varepsilon$ is a random error with mean zero

# One input variable vs Two





# Why estimate *f*

- Prediction
  - We can make accurate predictions for response Y for a new value of X
  - Example
    - Interested in predicting whether an individual will apply for a new credit card based on observations from 150, 000 people for whom we have information about over 100 attributes
    - Would like to know who should I send a credit card offer
- Inference
  - Interested in the type of relation between Y and X's
  - Which predictors affect the response? Is the relation positive or negative? Is the relation linear or more sophisticated?
  - Example
    - Predicting median house price based on different attributes

# How to estimate *f*

- We assume we have a set of training data

$$\{(X_1\ \ Y_1\ ),(X_2\ \ Y_2\ ),\ ....(X_n\ \ Y_n\ )\}$$

- Use training data to fit or train the model using statistical learning methods
  - Parametric methods
  - Non-parametric methods

# Parametric Methods

- Simplifies the problem of estimating ***f*** down to one of estimating a set of parameters.
- Utilize a two-step model-based approach
- Step1: Make assumption about the functional form of ***f***. The most common example is a linear model.

$$f(x) = \beta_0 + \beta_1\, x_1 + \beta_2\, x_2 \,\ldots\ldots + \beta_p\, x_p$$

- Step2: Use training data to fit the model aka estimate the unknown parameters $\beta_0, \beta_1, \ldots.\ \beta_p$

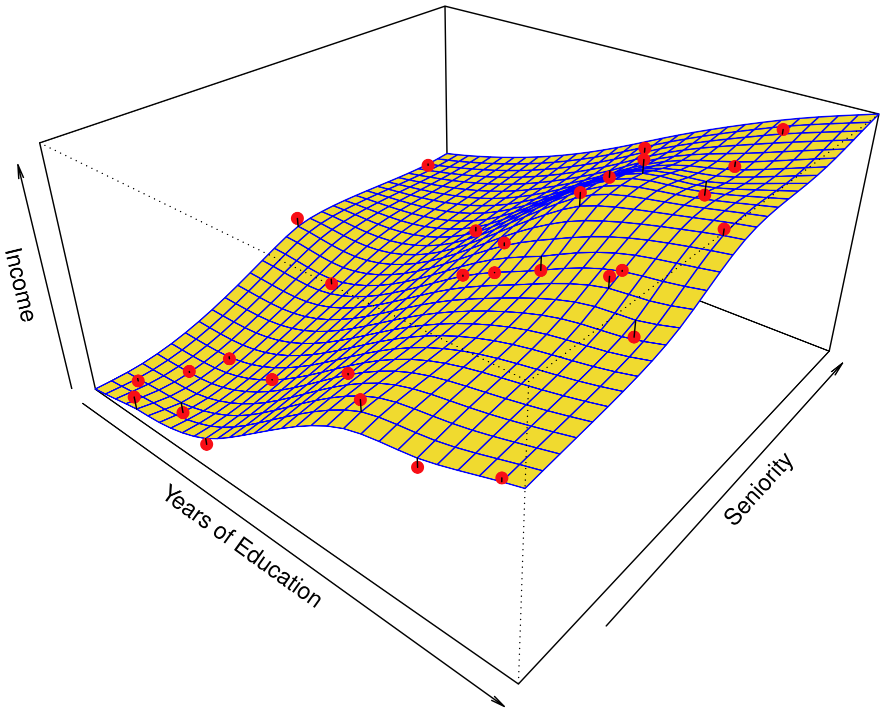
# Flexible Models

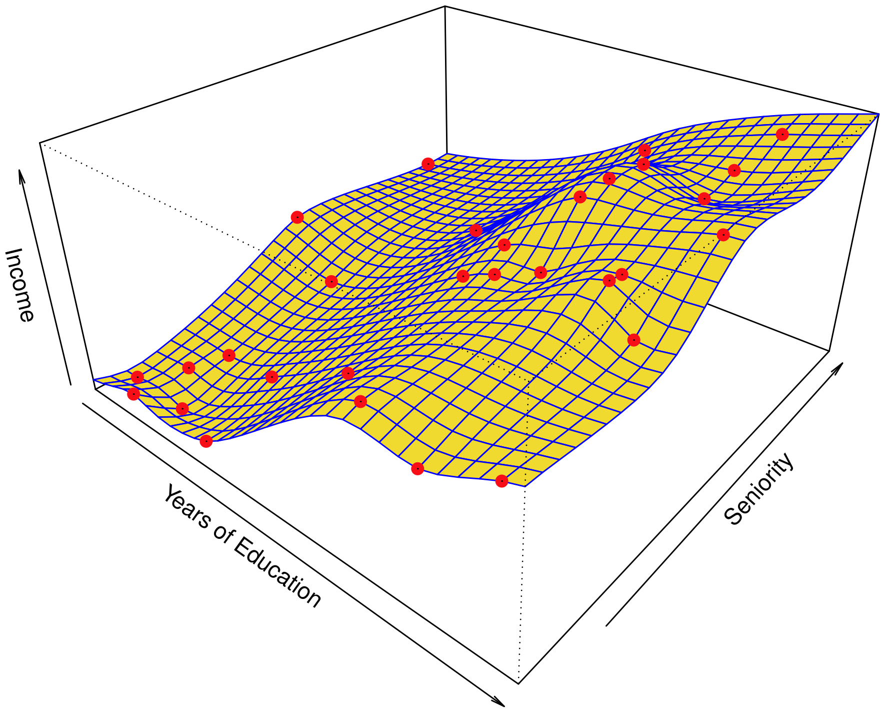
- In general, it is much simpler to estimate a set of parameters than it is to estimate an entirely arbitrary function ***f***. A disadvantage of this approach is that the specified model will not usually match the true form of ***f***.
- Using more flexible models is one way to attempt to combat inaccuracies in the chosen model. The more flexible the model, the more realistic it is. However, more flexible models have the disadvantage of requiring a greater number of parameters to be estimated and they are also more susceptible to overfitting.
- Overfitting is a phenomenon where a model closely matches the training data such that it captures too much of the noise or error in the data. This results in a model that fits the training data very well but does not make good predictions under test or in general.

# Non-Parametric Methods

- Do not make explicit assumptions about the functional form of ***f***.
- Hence, they fit a wider range of possible shapes of ***f***
- Very large number of observations are required to obtain accurate estimate of ***f***
- Example – Thin-Plate Spline
- Though less flexible, more restrictive models are more limited in the shapes they can estimate, they are easier to interpret because the relation of the predictors to the output is more easily understood.

# Non-Parametric Methods

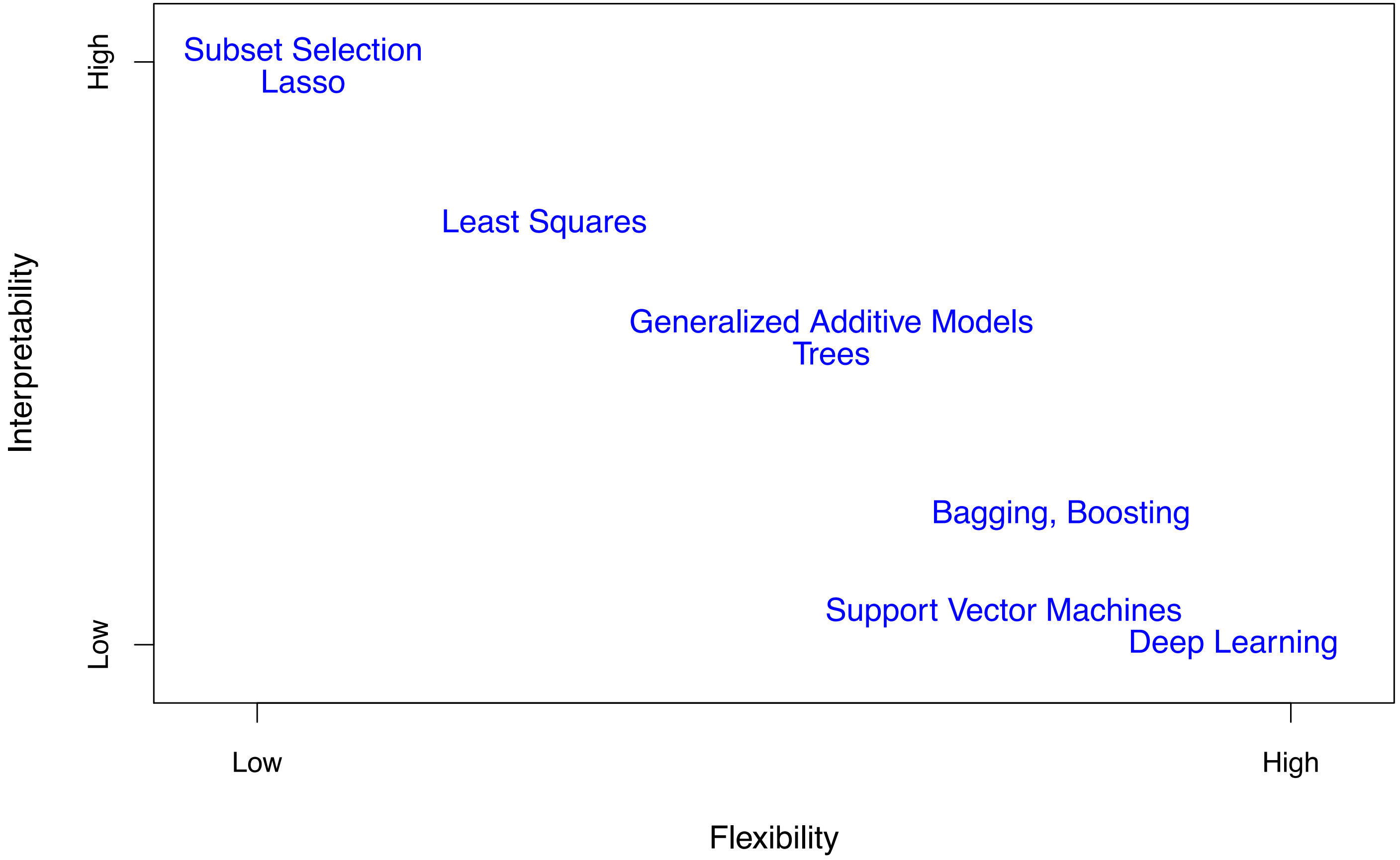




# Prediction Accuracy vs Model Interpretability

- Non-linear regression methods are more flexible and can potentially provide more accurate estimates.
- Why not just use a more flexible method if it is more realistic?
    - A simple method such as linear regression produces a model which is much easier to interpret (the Inference part is better). For example, in a linear model, $\beta_j$ is the average increase in Y for a one unit increase in $X_j$ holding all other variables constant.
    - Even if you are only interested in prediction, so the first reason is not relevant, it is often possible to get more accurate predictions with a simple, instead of a complicated, model. This seems counter intuitive but has to do with the fact that it is harder to fit a more flexible model.

# Prediction Accuracy vs Model Interpretability



# Supervised vs Unsupervised Learning

- Supervised learning is where both the predictors $X_i$ and the response variable $Y_i$ are observed. We generate a model that relates the predictor variables to the response variable with the goal of accurately predicting future observations or inferring the relation between the predictors and the response.
- Unsupervised learning refers to those situations where the $X_i$ are only observed and there is no associated response variable $Y_i$ . A common example is market segmentation where we try to segregate potential customers into groups based on their characteristics. Clustering is an example of unsupervised learning.

# Regression vs Classification

- Regression is used when $Y_i$ is a quantitative variable ( i.e., continuous or numerical)
  - Predicting the value of DOW in 6 months
  - Predicting the price of a house
- Classification is used when the response variable is categorical or qualitative i.e., it can take on values in one of K different classes.
  - Will the DOW go up or down in 6 months?
  - Is an email SPAM or not?

# Quality of Fit for Regression

- To evaluate the performance of a model, it is necessary to quantify how close the predicted responses are to the observed/actual data
- One common measure of accuracy in regression method is the mean squared error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Where, $\widehat{y_i}$ $is\ the\ predicted\ responses$

# Quality of Fit for Regression

- MSE will be small when the predicted responses are closer to the actual/true responses. With linear regression, we try to minimize the MSE on the training data
- MSE is also applied when testing a model and we really care about how well the model performs on the test data.
- Though it may be tempting to optimize the training mean squared error, the reality is that the model is judged by the accuracy of its predictions against unseen test data. As such, the model that yields the best test mean squared error is preferable to the model that yields the best training mean squared error.

# Training vs Test MSE

- There is no guarantee that the method with the smallest training MSE will have the smallest test (i.e., new data) MSE.
- In general, the more flexible a method is the lower its training MSE will be i.e., it will “fit” or explain the training data very well.
  - Side Note: More flexible methods (such as splines) can generate a wider range of possible shapes to estimate f as compared to less flexible and more restrictive methods (such as linear regression). The less flexible the method, the easier to interpret the model. Thus, there is a trade-off between flexibility and model interpretability.
- However, the test MSE may in fact be higher for a more flexible method than for a simple approach like linear regression.
- Overfitting refers specifically to scenarios in which a less flexible model would have yielded a smaller test mean squared error.

# Bias Variance Tradeoff

- Choice of learning method is governed by two competing factors – bias and variance
- Bias refers to the error introduced by modeling a usually extremely complicated problem using a simple problem
- For example – a linear regression model assumes a linear relation between Y and X which may be unlikely in real life thus introducing some bias
- More flexible (or complex) models have less bias

# Bias Variance Tradeoff

- Variance refers to the amount by which ***f*** would change if it were estimated with a different training set
- The more flexible a method is, greater is its variance

- In general, as the flexibility of the statistical method increases, its variance increases and bias decreases
- The relationship between bias, variance, and test set mean squared error is referred to as the bias-variance trade-off. It is called a trade-off because it is a challenge to find a model that has both a low variance and a low squared bias.

# Bias Variance Tradeoff

- For any given $x = x_{0,}$ the expected test mean squared error can be decomposed into the sum of the following three quantities:
  - Variance of $f(x_0)$
  - Squared bias of $f(x_0)$
  - Variance of the error term (***ε***)

$$Expected\,Test\,MSE = E\left(Y - f(x_0)\right)^2 = Bias^2 + Var + \underbrace{\sigma^2}_{\text{Irreducible Error}}$$

- To minimize the expected test error, it's necessary to choose a method that achieves both low variance and low bias. It can be seen that the expected test mean squared error can never be less than , the irreducible error.

# Accuracy for Classification Problems

- In classification scenarios, the most common means of quantifying the accuracy of is the training error rate. The training error rate is the proportion of errors that are made when applying to the training observations.

$$Error\ Rate = \sum_{i=1}^{n} I(y_i \neq \hat{y}_i) / n$$

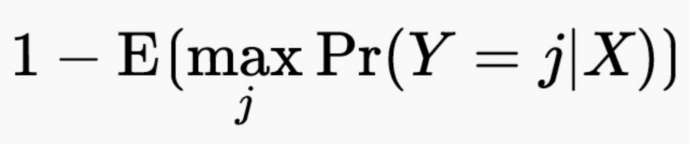
- $I(y_i \neq \hat{y}_i)$ is an indicator function, which will give 1 if the condition is correct, otherwise it gives a 0.
- Thus, the error rate represents the fraction of incorrect classifications, or misclassifications
- A good classifier is the one for which the test error rate is the smallest.

# Bayes Classifier

- It is possible to show that the test error rate is minimized on average by a very simple classifier that assigns each observation to the most likely class given its predictor variables.
- In Bayesian terms, a test observation should be classified for the predictor vector $x_0$, to the class j for which $\Pr(Y = j|X = x_0)$ is largest. That is, the class for which the conditional probability that Y = j, given the observed predictor vector $x_0$ is largest. This classifier is called the Bayes classifier.
- In a two-class scenario, this can be restated as $\Pr(Y = 1|X = x_0) > 0.5$ matching class A when TRUE and class B when FALSE .
- The threshold where the classification probability is exactly 50% is known as the Bayes decision boundary.

# Bayes Error Rate

- The Bayes error rate refers to the lowest possible error rate that could be achieved if somehow we knew exactly what the “true” probability distribution of the data looked like.



$$1 - \mathrm{E}\left(\max_{j} \Pr(Y = j|X)\right)$$

- On test data, no classifier (or stat. learning method) can get lower error rates than the Bayes error rate.
- Of course in real life problems the Bayes error rate can’t be calculated exactly.
- The Bayes error rate can also be described as the ratio of observations that lie on the "wrong" side of the decision boundary.
- Unfortunately, the conditional distribution of Y given X is often unknown, so the Bayes classifier is most often unattainable.

# K-Nearest Neighbors

- Many modeling techniques try to compute the conditional distribution of Y given X and then provide estimated classifications based on the highest estimated probability. The K nearest neighbors classifier is one such method.
- The K-nearest neighbors classifier takes a positive integer and first identifies the points that are nearest to $x_0$represented by $N_0$. It next estimates the conditional probability for class j based on the fraction of points in $N_0$ who have a response equal to j
- Formally, the estimated conditional probability can be stated as

$$\Pr(Y = j | X = x_0) = \frac{1}{k} \sum_{i \in N_0} \mathrm{I}(y_i = j)$$

- The K-Nearest Neighbor classifier then applies Bayes theorem and yields the classification with the highest probability.
- The choice of K can have a drastic effect on the yielded classifier. Too low a K yields a classifier that is too flexible, has too high a variance, and low bias.
- Conversely, as K increases, the yielded classifier becomes less flexible, with a low variance, but high bias.