Introduction to Data Science for Public Policy

Class 9: Introduction to Machine Learning

Thomas Monk

t.d.monk@lse.ac.uk

adapted from Introduction to Statistical Learning, (http://www.statlearning.com), and with thanks to Dr. Jack Blumenau, UCL.

September 9 2022

Overview of Core Concepts

Warning

Warning: this set of slides is pretty dense. I want to relate what we're doing to the theory of PP455 as much as possible.

Don't worry if you're not taking it all in right now - definitely ask me lots of questions as we go through.

Two main approaches to machine learning

- Supervised learning
- Unsupervised learning

The Supervised Learning Problem

PP455 introduced us to data from an **econometric** perspective. You will notice that all of the underlying concepts remain the same, we will simply speak about them slightly differently. Starting point:

- Outcome measurement Y (also called dependent variable, response, target).

The Supervised Learning Problem

PP455 introduced us to data from an **econometric** perspective. You will notice that all of the underlying concepts remain the same, we will simply speak about them slightly differently. Starting point:

- Outcome measurement Y (also called dependent variable, response, target).
- Vector of p predictor measurements X (also called inputs, regressors, covariates, features, independent variables).

The Supervised Learning Problem

PP455 introduced us to data from an **econometric** perspective. You will notice that all of the underlying concepts remain the same, we will simply speak about them slightly differently. Starting point:

- Outcome measurement Y (also called dependent variable, response, target).
- Vector of p predictor measurements X (also called inputs, regressors, covariates, features, independent variables).
- In the regression problem, Y is quantitative (e.g price, blood pressure).

The Supervised Learning Problem

PP455 introduced us to data from an **econometric** perspective. You will notice that all of the underlying concepts remain the same, we will simply speak about them slightly differently. Starting point:

- Outcome measurement Y (also called dependent variable, response, target).
- Vector of p predictor measurements X (also called inputs, regressors, covariates, features, independent variables).
- In the regression problem, Y is quantitative (e.g price, blood pressure).
- In the classification problem, Y takes values in a finite, unordered set (survived/died, digit 0-9, cancer class of tissue sample).

The Supervised Learning Problem

PP455 introduced us to data from an **econometric** perspective. You will notice that all of the underlying concepts remain the same, we will simply speak about them slightly differently. Starting point:

- Outcome measurement Y (also called dependent variable, response, target).
- Vector of p predictor measurements X (also called inputs, regressors, covariates, features, independent variables).
- In the regression problem, Y is quantitative (e.g price, blood pressure).
- In the classification problem, Y takes values in a finite, unordered set (survived/died, digit 0-9, cancer class of tissue sample).
- We have training data $(x_1, y_1), ..., (x_N, y_N)$. These are observations (examples, instances) of these measurements.

Objectives

On the basis of the training data we would like to: Starting point:

- Accurately predict unseen test cases.

Objectives

On the basis of the training data we would like to: Starting point:

- Accurately predict unseen test cases.
- Understand which inputs affect the outcome, and how.

Objectives

On the basis of the training data we would like to: Starting point:

- Accurately predict unseen test cases.
- Understand which inputs affect the outcome, and how.
- Assess the quality of our predictions and inferences.

Objectives

On the basis of the training data we would like to: Starting point:

- Accurately predict unseen test cases.
- Understand which inputs affect the outcome, and how.
- Assess the quality of our predictions and inferences.

Objectives

On the basis of the training data we would like to: Starting point:

- Accurately predict unseen test cases.
- Understand which inputs affect the outcome, and how.
- Assess the quality of our predictions and inferences.

Note that causality is not necessarily one of the objectives, as we discussed in the first class.

Unsupervised Learning

- No outcome variable, just a set of predictors (features) measured on a set of samples.

Unsupervised Learning

- No outcome variable, just a set of predictors (features) measured on a set of samples.
- Objective is more fuzzy – i.e. find groups of samples that behave similarly, find features that behave similarly, find linear combinations of features with the most variation.

Unsupervised Learning

- No outcome variable, just a set of predictors (features) measured on a set of samples.
- Objective is more fuzzy – i.e. find groups of samples that behave similarly, find features that behave similarly, find linear combinations of features with the most variation.
- Difficult to know how well you are doing. Different from supervised learning, but can be useful as a pre-processing step for supervised learning.

Unsupervised Learning

- No outcome variable, just a set of predictors (features) measured on a set of samples.
- Objective is more fuzzy – i.e. find groups of samples that behave similarly, find features that behave similarly, find linear combinations of features with the most variation.
- Difficult to know how well you are doing. Different from supervised learning, but can be useful as a pre-processing step for supervised learning.
- We'll not discuss, but I'll show some examples.

Machine learning

Machine learning

- Machine learning refers to a vast set of tools for understanding data.

Machine learning

- Machine learning refers to a vast set of tools for understanding data.
- For a quantitative response Y and a set of predictors X:

$$Y = f(X) + \epsilon$$

Machine learning

- Machine learning refers to a vast set of tools for understanding data.
- For a quantitative response Y and a set of predictors X:

$$Y = f(X) + \epsilon$$

- Here, f represents the systematic information that X provides about Y - i.e. an arbitrary function.

Machine learning

- Machine learning refers to a vast set of tools for understanding data.
- For a quantitative response Y and a set of predictors X:

$$Y = f(X) + \epsilon$$

- Here, f represents the systematic information that X provides about Y - i.e. an arbitrary function.
- Statistical learning refers to a set of approaches for estimating f.

Machine learning

- Machine learning refers to a vast set of tools for understanding data.
- For a quantitative response Y and a set of predictors X:

$$Y = f(X) + \epsilon$$

- Here, f represents the systematic information that X provides about Y - i.e. an arbitrary function.
- Statistical learning refers to a set of approaches for estimating f.
- What method(s) of estimating f have we **already seen**?

Machine learning

- Machine learning refers to a vast set of tools for understanding data.
- For a quantitative response Y and a set of predictors X:

$$Y = f(X) + \epsilon$$

- Here, f represents the systematic information that X provides about Y - i.e. an arbitrary function.
- Statistical learning refers to a set of approaches for estimating f.
- What method(s) of estimating f have we **already seen**?
  - **OLS**! $Y = \beta_0 + \sum_{i=1}^{N} \beta_1 X_i + \epsilon$

Machine learning

- Machine learning refers to a vast set of tools for understanding data.
- For a quantitative response Y and a set of predictors X:

$$Y = f(X) + \epsilon$$

- Here, f represents the systematic information that X provides about Y - i.e. an arbitrary function.
- Statistical learning refers to a set of approaches for estimating f.
- What method(s) of estimating f have we **already seen**?
  - **OLS**! $Y = \beta_0 + \sum_{i=1}^{N} \beta_1 X_i + \epsilon$
  - **Logit/Probit**! $Y = \tilde{f}(\sum_{i=1}^{N} \beta_1 X_i) + \epsilon$

Machine learning

- Machine learning refers to a vast set of tools for understanding data.
- For a quantitative response Y and a set of predictors X:

$$Y = f(X) + \epsilon$$

- Here, f represents the systematic information that X provides about Y - i.e. an arbitrary function.
- Statistical learning refers to a set of approaches for estimating f.
- What method(s) of estimating f have we **already seen**?
  - **OLS**! $Y = \beta_0 + \sum_{i=1}^{N} \beta_1 X_i + \epsilon$
  - **Logit/Probit**! $Y = \tilde{f}(\sum_{i=1}^{N} \beta_1 X_i) + \epsilon$
  - Finding the solution to these (i.e estimating $\hat{\beta}$) simply minimises the mean squared error - we've done this many, many times!

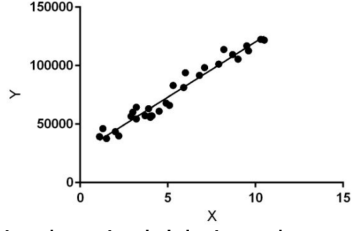
OLS as ‘machine learning’

ML | Linear Regression

Difficulty Level : Medium • Last Updated : 22 Aug, 2022

Read Discuss

Linear Regression is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.



You have already done machine learning! It’s just that now things are framed differently.

What's changed?

- In our move from causal **inference** to **prediction**, we will care much more about the **residual**.

What's changed?

- In our move from causal **inference** to **prediction**, we will care much more about the **residual**.
- The sum of the squared residual, averaged across observations is called **mean squared error**.

$$MSE_{training} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}\hat{\epsilon}^2$$

What's changed?

- In our move from causal **inference** to **prediction**, we will care much more about the **residual**.
- The sum of the squared residual, averaged across observations is called **mean squared error**.

$$MSE_{training} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}\hat{\epsilon}^2$$

- Why squared?

What's changed?

- In our move from causal **inference** to **prediction**, we will care much more about the **residual**.
- The sum of the squared residual, averaged across observations is called **mean squared error**.

$$MSE_{training} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}\hat{\epsilon}^2$$

- Why squared?

What's changed?

- In our move from causal **inference** to **prediction**, we will care much more about the **residual**.
- The sum of the squared residual, averaged across observations is called **mean squared error**.

$$MSE_{training} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}\hat{\epsilon}^2$$

- Why squared? As we care about **distance** from y_i, not the sign of the direction.

What's changed?

1. Prediction: $\hat{Y} = \hat{f}(X)$, where $\hat{f}$ is a black box.

What's changed?

1. Prediction: $\hat{Y} = \hat{f}(X)$, where $\hat{f}$ is a black box.
2. Inference: How Y changes as a function of X.

What's changed?

1. Prediction: $\hat{Y} = \hat{f}(X)$, where $\hat{f}$ is a black box.
2. Inference: How Y changes as a function of X.

- Depending on whether the ultimate goal is prediction, inference or a mix of both, we may deploy different methods for estimating f .

What's changed?

1. Prediction: $\hat{Y} = \hat{f}(X)$, where $\hat{f}$ is a black box.
2. Inference: How Y changes as a function of X.

- Depending on whether the ultimate goal is prediction, inference or a mix of both, we may deploy different methods for estimating f .
- Also depending on the ultimate goal you may or may not care about evaluating the causal relationship between Y and X

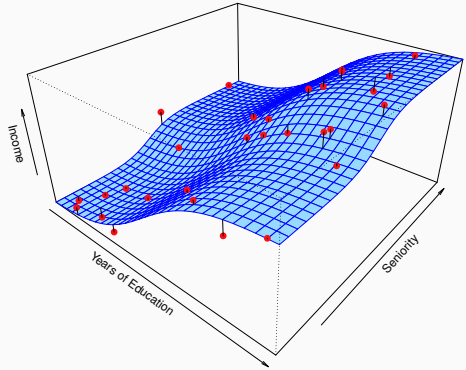
What's changed? - how we understand the error

- Consider the ideal predictor of Y, with regard to mean squared error: $f(x) = E(Y|X = x)$ is the function that minimises $E[(Y - g(X))^2|X = x]$ over all functions g at all points $X = x$.

What's changed? - how we understand the error

- Consider the ideal predictor of Y, with regard to mean squared error: $f(x) = E(Y|X = x)$ is the function that minimises $E[(Y - g(X))^2|X = x]$ over all functions g at all points $X = x$.
- $\varepsilon \equiv Y - f(x)$ is the **irreducible** error - i.e. even if we know exactly what $f(x)$ is we would still make errors in prediction, since there is an underlying assumption that at each $X = x$ there is typically a distribution of possible Y values.

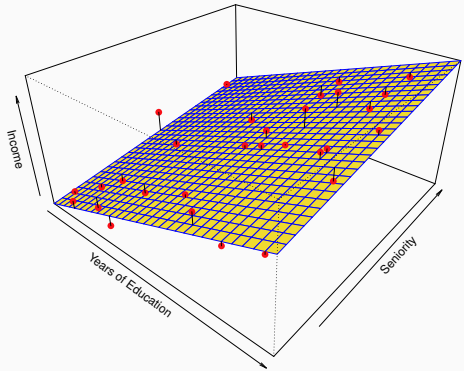
Simulated example



Red points are simulated values for `income` from the model

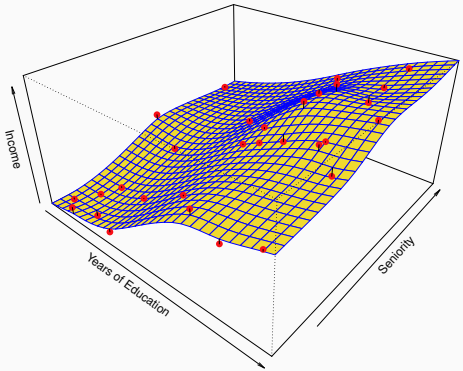
$$income = f(education, seniority) + \epsilon$$

f is the blue surface.

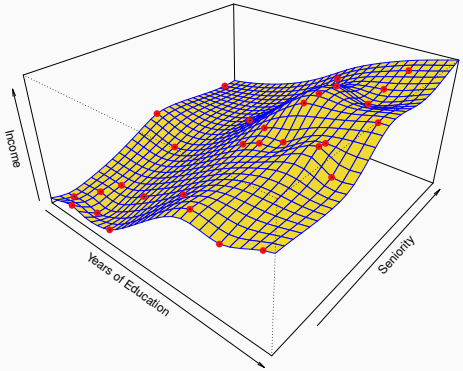


Linear regression model fit to the simulated data.

$$\hat{f}_L(education, seniority) = \hat{\beta}_0 + \hat{\beta}_1 \times education + \hat{\beta}_2 \times seniority$$



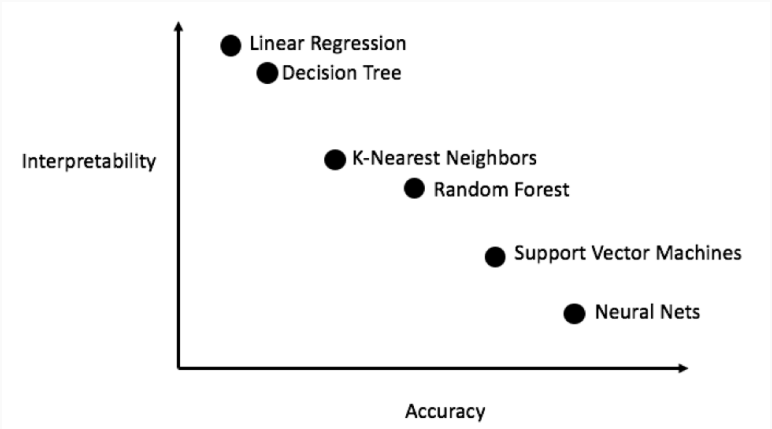
- More flexible regression model $\hat{f}_S(education, seniority)$ fit to the simulated data.
- Here we use a technique called a thin-plate spline to fit a flexible surface.
- We control the roughness of the fit.



- Even more flexible spline regression model $\hat{f}_S(education, seniority)$ fit to the simulated data.
- Here the fitted model makes no errors on the training data!
- Also known as overfitting.

Some trade-offs

- Prediction accuracy versus interpretability.
  - Linear models are easy to interpret; thin-plate splines are not.
- Good fit versus over-fit or under-fit.
  - How do we know when the fit is just right?
- Parsimony versus black-box.
  - We often prefer a simpler model involving fewer variables over a black-box predictor involving them all.



Source:(https://medium.com/ansaro-blog/interpreting-machine-learning-models-1234d735d6c9)

Assessing model accuracy

- We discussed the distinction between **test** data and **training.**

Assessing model accuracy

- We discussed the distinction between **test** data and **training.**
- Test data is withheld from the data set, and the model is *only* fit on the training data, T_r.

Assessing model accuracy

- We discussed the distinction between **test** data and **training.**
- Test data is withheld from the data set, and the model is *only* fit on the training data, T_r.
- Our measure of model accuracy is by the results on the **test data only**, T_e, i.e:

$$MSE_{T_e} = Ave_{i \in T_e}[y_i - \hat{f}(x_i)]^2$$

Assessing model accuracy

- We discussed the distinction between **test** data and **training.**
- Test data is withheld from the data set, and the model is *only* fit on the training data, T_r.
- Our measure of model accuracy is by the results on the **test data only**, T_e, i.e:

$$MSE_{T_e} = Ave_{i \in T_e}[y_i - \hat{f}(x_i)]^2$$

- The above simply states the mean squared error of the training data is the average, across all items of the training data, of the distance between the training Y values and our prediction for them, $\hat{f}(x_i)$.

Assessing model accuracy

- We discussed the distinction between **test** data and **training.**
- Test data is withheld from the data set, and the model is *only* fit on the training data, T_r.
- Our measure of model accuracy is by the results on the **test data only**, T_e, i.e:

$$MSE_{T_e} = Ave_{i \in T_e}[y_i - \hat{f}(x_i)]^2$$

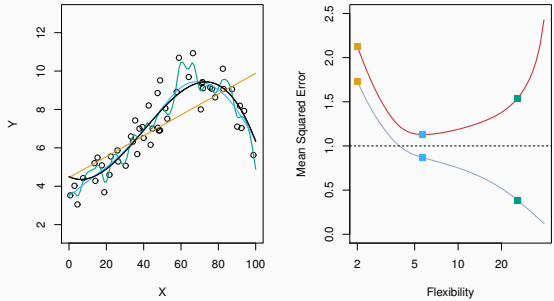
- The above simply states the mean squared error of the training data is the average, across all items of the training data, of the distance between the training Y values and our prediction for them, $\hat{f}(x_i)$.
- If we don't withhold data, we have **overfitting** problems as above. What do I mean by this?

Assessing model accuracy

- We discussed the distinction between **test** data and **training.**
- Test data is withheld from the data set, and the model is *only* fit on the training data, T_r.
- Our measure of model accuracy is by the results on the **test data only**, T_e, i.e:

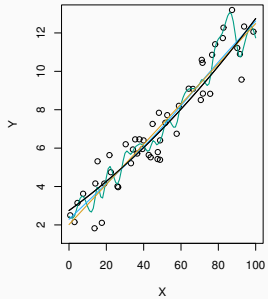
$$MSE_{T_e} = Ave_{i \in T_e}[y_i - \hat{f}(x_i)]^2$$

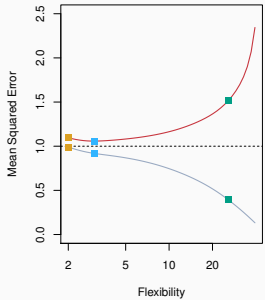
- The above simply states the mean squared error of the training data is the average, across all items of the training data, of the distance between the training Y values and our prediction for them, $\hat{f}(x_i)$.
- If we don't withhold data, we have **overfitting** problems as above. What do I mean by this?
- That our model works really well for the data we have **only**, and is of no use in the real world.



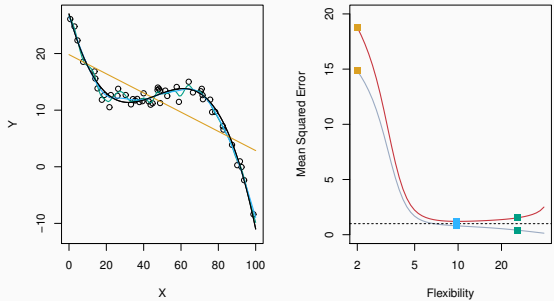
Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing splines.

- Black curve is truth.
- Red curve on right is MSE_{Te}, grey curve is MSE_{Tr}.
- Orange, blue and green curves/squares correspond to fits of different flexibility.





- The setup as before, using a different true f that is much closer to linear. In this setting, linear regression provides a very good fit to the data.
- Here the truth is smoother, so the smoother fit and linear model do really well.



- Setup as above, using a different f that is far from linear.
- In this setting, linear regression provides a very poor fit to the data.
- Here the truth is wiggly and the noise is low, so the more flexible fits do the best.

Classification problem

Here the response variable Y is **qualitative** - e.g. email is one of $\mathbb{C} = (spam; ham)(ham = goodemail)$, digitclass, if trying to classify handwritten digits, is one of $\mathbb{C} = \{0, 1, ..., 9\}$. Our goals are to:

- Build a classifier C(X) that assigns a class label from C to a future unlabelled observation X.

Classification problem

Here the response variable Y is **qualitative** - e.g. email is one of $\mathbb{C} = (spam; ham)(ham = goodemail)$, digitclass, if trying to classify handwritten digits, is one of $\mathbb{C} = \{0, 1, ..., 9\}$. Our goals are to:

- Build a classifier C(X) that assigns a class label from C to a future unlabelled observation X.
- Assess the uncertainty in each classification.

Classification problem

Here the response variable Y is **qualitative** - e.g. email is one of $\mathbb{C} = (spam; ham)(ham = goodemail)$, digitclass, if trying to classify handwritten digits, is one of $\mathbb{C} = \{0, 1, ..., 9\}$. Our goals are to:

- Build a classifier C(X) that assigns a class label from C to a future unlabelled observation X.
- Assess the uncertainty in each classification.
- Understand the roles of the different predictors among $(X_1, X_2, ..., X_p$

Classification problem

Here the response variable Y is **qualitative** - e.g. email is one of $\mathbb{C} = (spam; ham)(ham = goodemail)$, digitclass, if trying to classify handwritten digits, is one of $\mathbb{C} = \{0, 1, ..., 9\}$. Our goals are to:

- Build a classifier C(X) that assigns a class label from C to a future unlabelled observation X.
- Assess the uncertainty in each classification.
- Understand the roles of the different predictors among $(X_1, X_2, ..., X_p$

Classification problem

Here the response variable Y is **qualitative** - e.g. email is one of $\mathbb{C} = (spam; ham)(ham = goodemail)$, digitclass, if trying to classify handwritten digits, is one of $\mathbb{C} = \{0, 1, ..., 9\}$. Our goals are to:

- Build a classifier C(X) that assigns a class label from C to a future unlabelled observation X.
- Assess the uncertainty in each classification.
- Understand the roles of the different predictors among $(X_1, X_2, ..., X_p$

Recall from PP455, we can let:

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

that is, the probability of Y being in a certain class, given all of our features X.

Estimating $p_k(x)$

- We saw that the linear probability model does exactly this in PP455!

Estimating $p_k(x)$

- We saw that the linear probability model does exactly this in PP455!
- We can also use Logit or Probit.

Estimating $p_k(x)$

- We saw that the linear probability model does exactly this in PP455!
- We can also use Logit or Probit.
- Another estimation technique used is K-nearest neighbour (we can also use this in 'regression').

Estimating $p_k(x)$

- We saw that the linear probability model does exactly this in PP455!
- We can also use Logit or Probit.
- Another estimation technique used is K-nearest neighbour (we can also use this in 'regression').
  - Given any values for X's, find the nearest K $X's$ to them in our training data set.

Estimating $p_k(x)$

- We saw that the linear probability model does exactly this in PP455!
- We can also use Logit or Probit.
- Another estimation technique used is K-nearest neighbour (we can also use this in 'regression').
  - Given any values for X's, find the nearest K $X's$ to them in our training data set.
  - **Vote** for the class that our prediction should take, based on what class is assigned to those closest K X values.

Estimating $p_k(x)$

- We saw that the linear probability model does exactly this in PP455!
- We can also use Logit or Probit.
- Another estimation technique used is K-nearest neighbour (we can also use this in 'regression').
  - Given any values for X's, find the nearest K $X's$ to them in our training data set.
  - **Vote** for the class that our prediction should take, based on what class is assigned to those closest K X values.
  - Assign the class of our prediction based on those votes.

Estimating $p_k(x)$

- We saw that the linear probability model does exactly this in PP455!
- We can also use Logit or Probit.
- Another estimation technique used is K-nearest neighbour (we can also use this in 'regression').
  - Given any values for X's, find the nearest K $X's$ to them in our training data set.
  - **Vote** for the class that our prediction should take, based on what class is assigned to those closest K X values.
  - Assign the class of our prediction based on those votes.
  - Draw a diagram here?

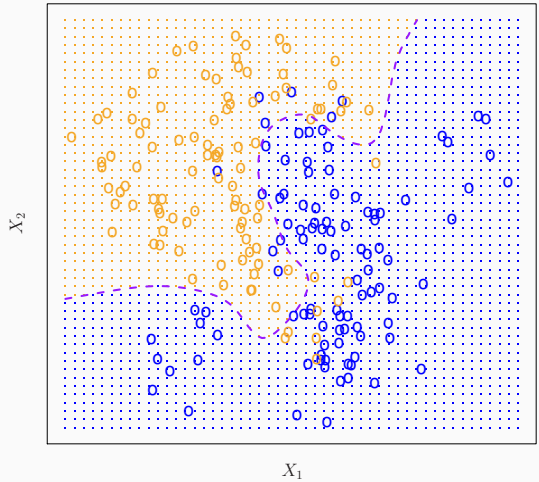
Estimating $p_k(x)$

- We saw that the linear probability model does exactly this in PP455!
- We can also use Logit or Probit.
- Another estimation technique used is K-nearest neighbour (we can also use this in 'regression').
  - Given any values for X's, find the nearest K $X's$ to them in our training data set.
  - **Vote** for the class that our prediction should take, based on what class is assigned to those closest K X values.
  - Assign the class of our prediction based on those votes.
  - Draw a diagram here?
  - Performance dramatically decreases as P (number of features) increases - our set becomes **sparse**, with not very much data around our new point to help with prediction.

Estimating $p_k(x)$

- We saw that the linear probability model does exactly this in PP455!
- We can also use Logit or Probit.
- Another estimation technique used is K-nearest neighbour (we can also use this in 'regression').
  - Given any values for X's, find the nearest K $X's$ to them in our training data set.
  - **Vote** for the class that our prediction should take, based on what class is assigned to those closest K X values.
  - Assign the class of our prediction based on those votes.
  - Draw a diagram here?
  - Performance dramatically decreases as P (number of features) increases - our set becomes **sparse**, with not very much data around our new point to help with prediction.
- Example in next slide: class is either Orange or Blue. We classify based on two features: X_1 and X_2.

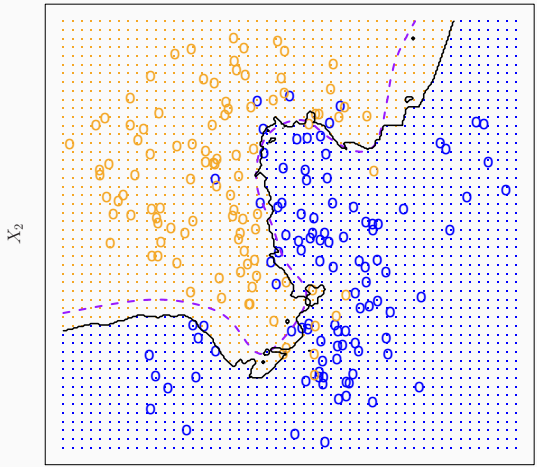
Example: K-nearest neighbors in two dimensions



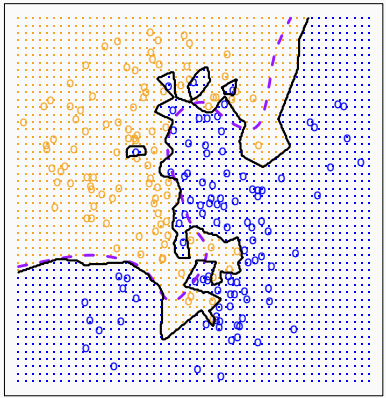
Note something interesting purple line is the Bayes decision boundary - the best we can do, with zero irreducible error. Why not perfect?

Note something interesting purple line is the Bayes decision boundary - the best we can do, with zero irreducible error. Why not perfect?Because of overlap between the groups!

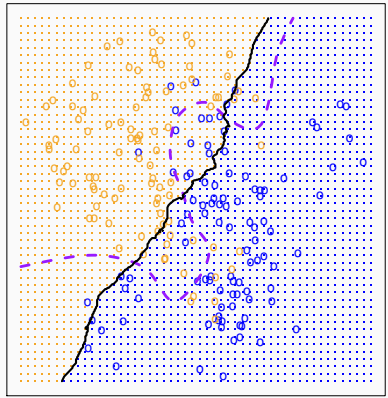
KNN: K=10

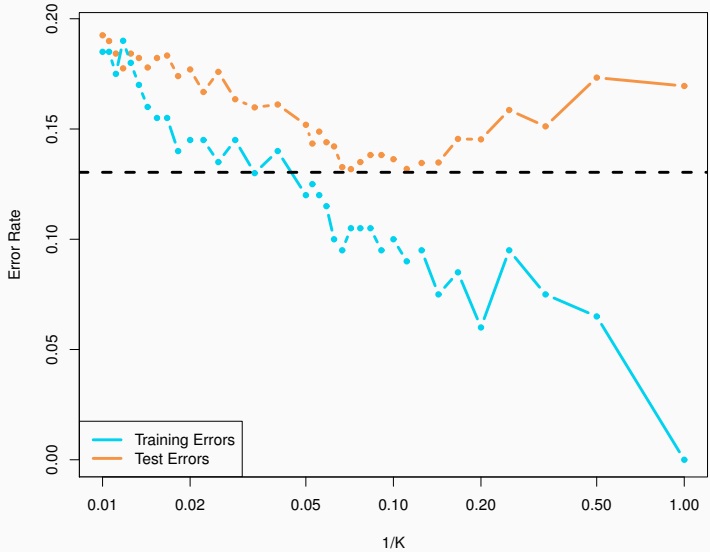


KNN: K=1



KNN: K=100




Error Rate
0.00
0.05
0.10
0.15
0.20
0.01
0.02
0.05
0.10
0.20
0.50
1.00
1/K
Training Errors
Test Errors

Enter the Kaggle house prices compeition.