Agenda

Start	End	Item
		Logistic Regression
		Break
		East Side Vs West Side!
		Absenteeism KNN example

Supervised Learning

Inferring a function from labeled data.

“Learn from telling”, “Look at my data and I will tell you what to predict”

Business Context

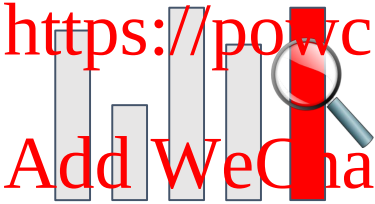
Marketing-Will a customer buy yes or no? How much will a customer spend?

Operations- Will an applicant default? When will a machine break?

Sports Analytics- How many points will the Bears’ QB score? What is the Bears’ probability of winning?

Requires expertise and stakeholder buy in

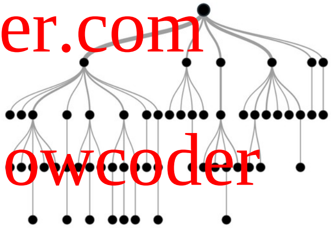
Data Setup



Flat “Excel” file. Each row is a record or observation. Each column is an attribute of the record.

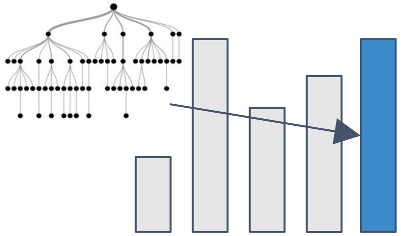
One column is the outcome, y or target attribute.

Algorithm



Modeling e.g. K-NN, linear regression, decision tree, random forest etc.

Application



Use the model to make predictions for the target label on the new data.

Logistic Regression

- Extends idea of linear regression to situation where outcome variable is categorical

- Instead of ordinary least squares, β are derived through an iterative process called *maximum likelihood estimation*

- We focus on binary classification
  i.e. Y=0 or Y=1

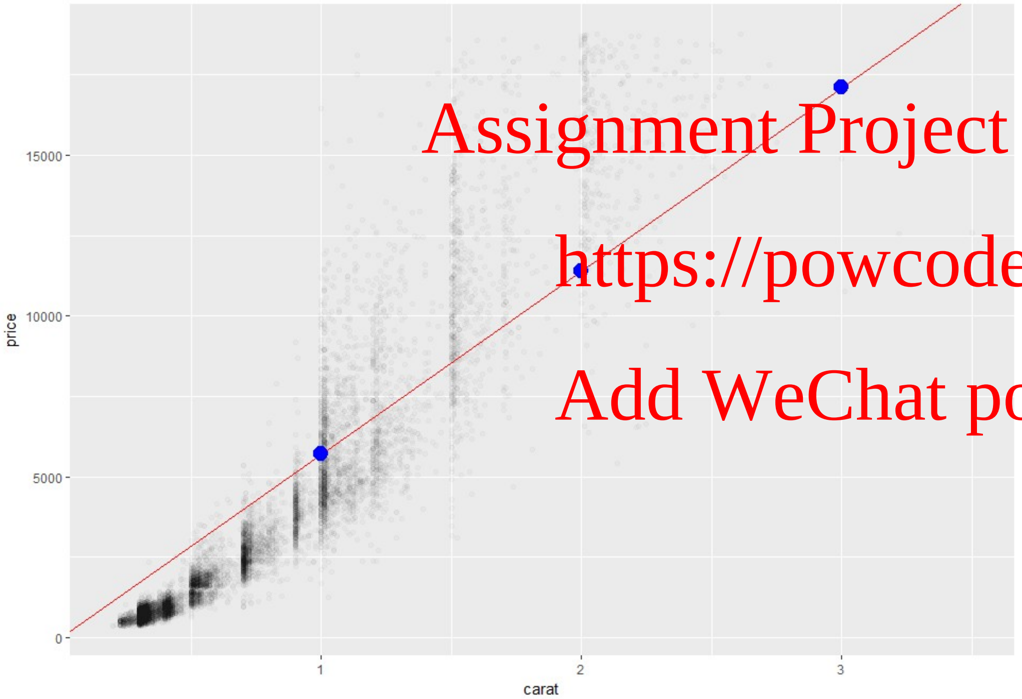
Regression Equation Review

Regression

How Many Cones?

= + *temperature + *day + *price + error

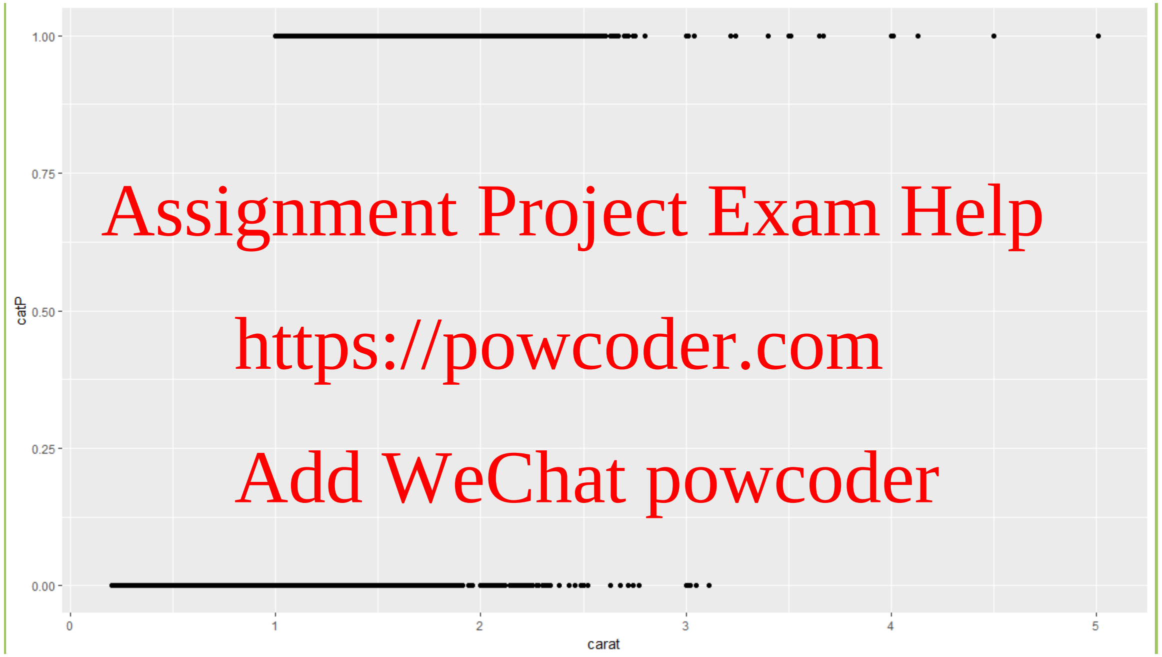
Linear Regression



- The predictions is continuous…it continues forever.

A binary relationship between carat and price

Diamonds above or below $11K



Step 1: Logistic Response Function

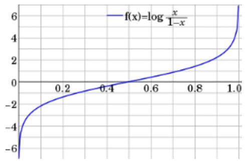
Regression | How Many Cones?

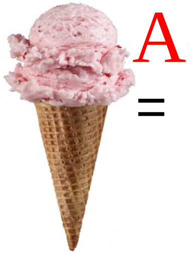
#

= + *temperature + *day + *price + error

Logistic Regression | Will they buy a cone Y/N?

Logit of





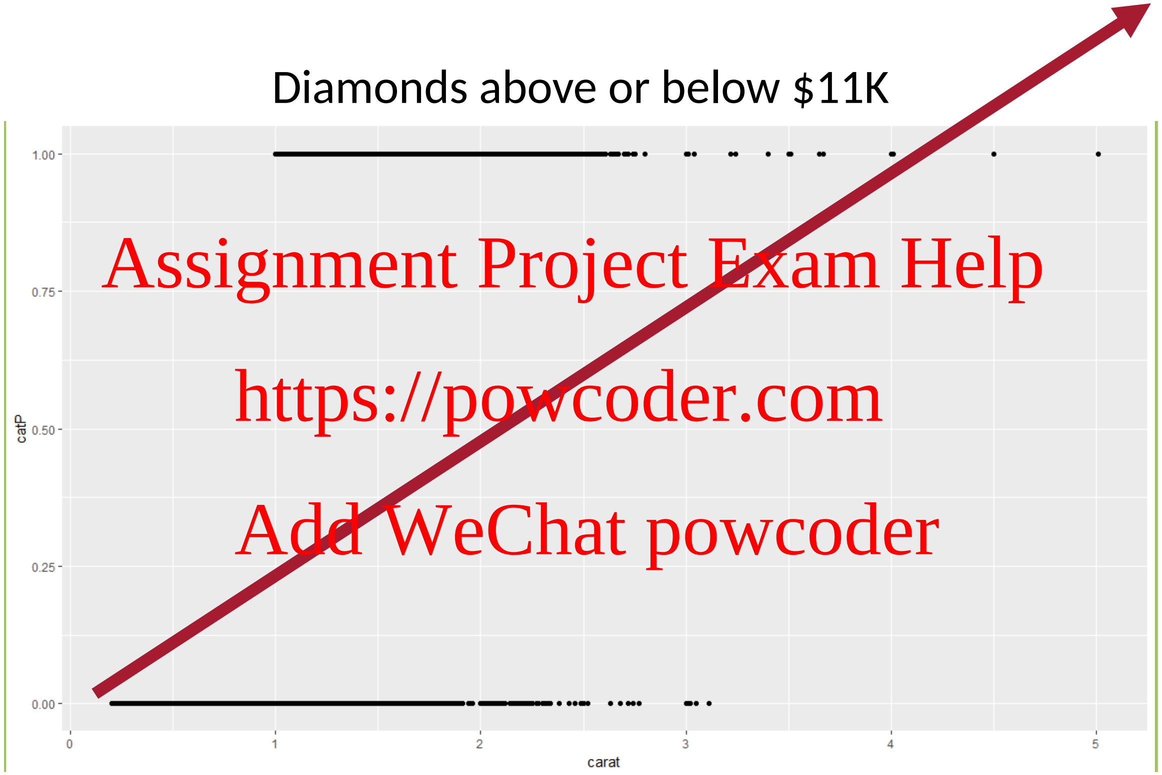
= + *temperature + *day + *price + error

We will let R handle calculating the equation output logOdds to the more understandable probability.

Let’s see the difference in practice

Open A_lm_for classes.R

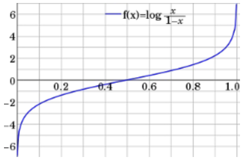
A binary relationship between carat and price



If the data only has two values, 0/1 but the regression equation goes to infinity.

This makes no sense!

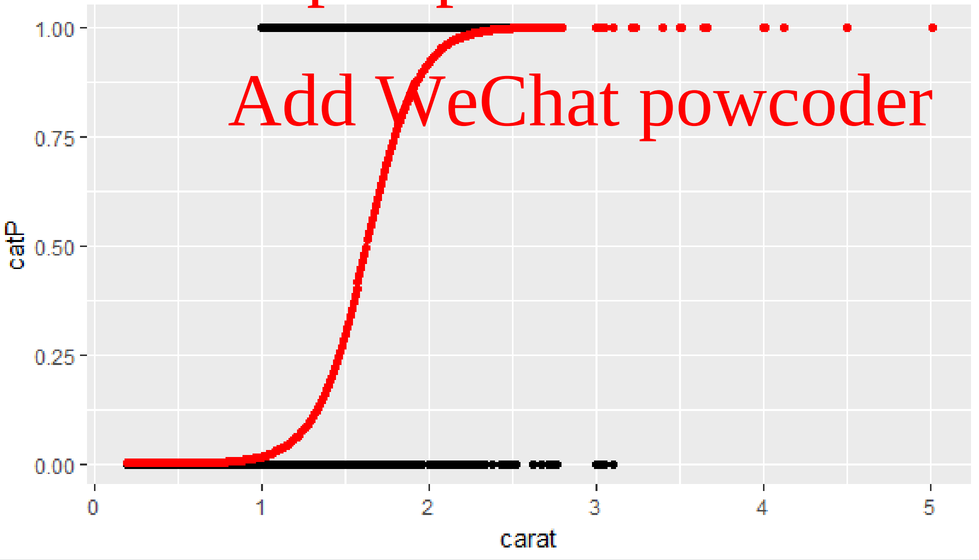
“is this diamond worth more than $11K or not.” Predicting 2 means 2 yes’es?

1. 

What is the log-odds of the price above $11K? = Beta + Beta*Carat

2. Convert to **probability** with logistic response function (e^l / (1+e^l)

3. The probabilities are more intuitive than the log-odds from the equation.



From probability to class, define a cutoff threshold.

- 0.50 is popular initial choice

- Additional considerations (see Chapter 5)
  - **Maximize classification accuracy**
  - Maximize sensitivity (subject to min. level of specificity)
  - Minimize false positives (subject to max. false negative rate)
  - Minimize expected cost of misclassification (need to specify costs)

If a team has a probability of .25 classify them as a loser.
If a team has .50 or more classify them as a winner

NCAA Classification Madness

- College Basketball
- Annual 64 team tournament

Business Impact:

- $1B wagered
- $2B in lost productivity
- Bragging Rights

Objective: Identify the probability of a team winning in Round 1.

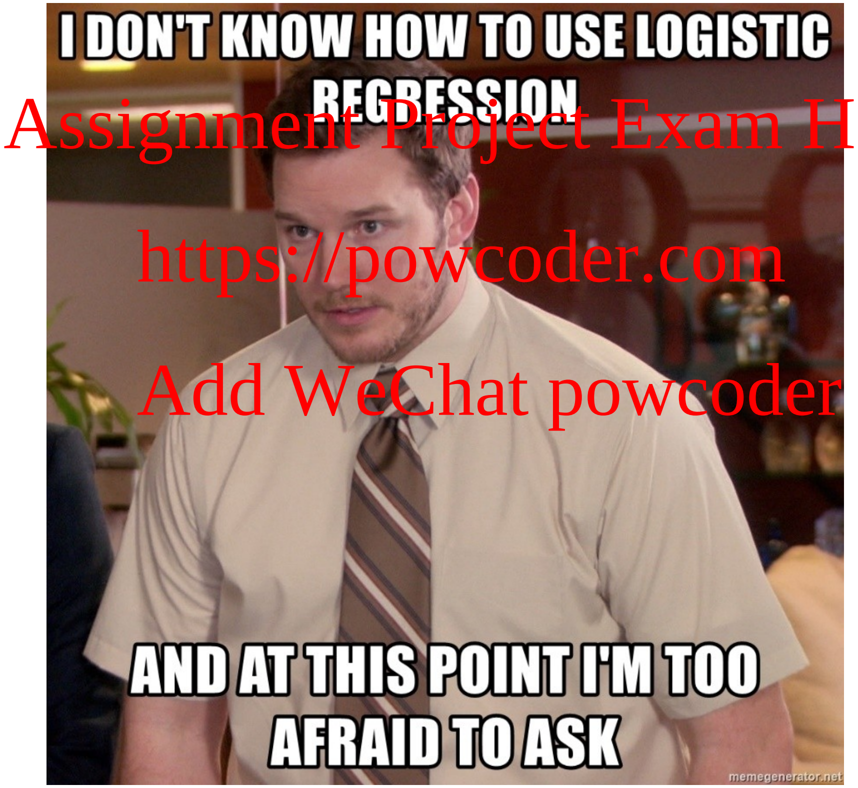
My friend Mandy is next level.

FiveThirtyEight: Hacking the bracket
Akamai Adaptive Media Player
AMP Premier - ABC News v2.88.8
FiveThirtyEight
00:00 / 05:02

https://fivethirtyeight.com/features/how-a-data-scientist-whod-never-heard-of-basketball-mastered-march-madness/

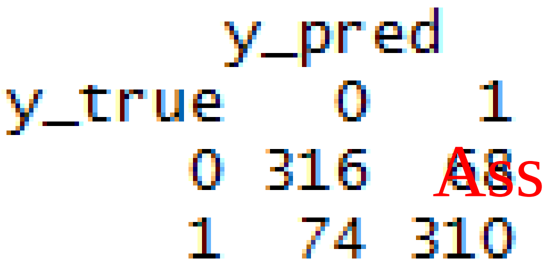
Let’s practice

Open B_fullyMarchMadnessREVISED.R


I DON'T KNOW HOW TO USE LOGISTIC REGRESSION
AND AT THIS POINT I'M TOO AFRAID TO ASK
memegenerator.net

Evaluating a Classification

Confusion Matrix



```
      y_pred
y_true   0   1
     0 316  68
     1  74 310
```

- The model predicted losers 390 (316 + 74) times
- The model was correct 316 times for losers
- The model predicted 378 winners (68+310)
- The model was correct 310 times for winners

As you progress in your data science education, learning other KPI (Recall, Precision, AUC etc) in Chapter 5 is worthwhile. In this course we stick with the basic accuracy.

The confusion matrix

	Actual 1	Actual 0
Predicted 1	True Positives	False Positives
Predicted 0	False Negatives	True Negatives

Probabilities are 0-1 so a “cutoff threshold” is used to classify into 1 or 0 in the matrix.

The confusion matrix

Actual	Probability
1	.45
1	.55
0	.95
1	.75
0	.25

Cutoff 0.01

	Actual 1	Actual 0
Predicted 1	3	2
Predicted 0	0	0

Cutoff 0.50

	Actual 1	Actual 0
Predicted 1	2	1
Predicted 0	1	1

Cutoff 0.75

	Actual 1	Actual 0
Predicted 1	1	1
Predicted 0	2	1

Cutoff 0.99

	Actual 1	Actual 0
Predicted 1	0	0
Predicted 0	3	2

Adjusting the cutoff impacts the numbers in the confusion matrix.

True/False Positive Rates

	Actual 1	Actual 0
Predicted 1	True Positives	False Positives
Predicted 0	False Negatives	True Negatives
	TruePosRate	FalsePosRate

Sensitivity or
True Positive Rate = TruePos / (TruePos + FalseNeg)

This is the proportion of the correct "1" classifications among all "1" actuals

Specificity or
False Positive Rate = FalsePos / (FalsePos + TrueNeg)

This is the proportion of the correct "0" classifications among all "0" actuals.

Plotting the different cutoff thresholds in a fake example

0.5 Cutoff

	Actual 1	Actual 0
Predicted 1	2	1
Predicted 0	1	1

● True Positive Rate = 2/ 3
False Positive Rate = 1/2

More balanced, optimizing accuracy

0.99 Cutoff

	Actual 1	Actual 0
Predicted 1	0	0
Predicted 0	3	2

● True Positive Rate = 0/ 3
False Positive Rate = 0/2

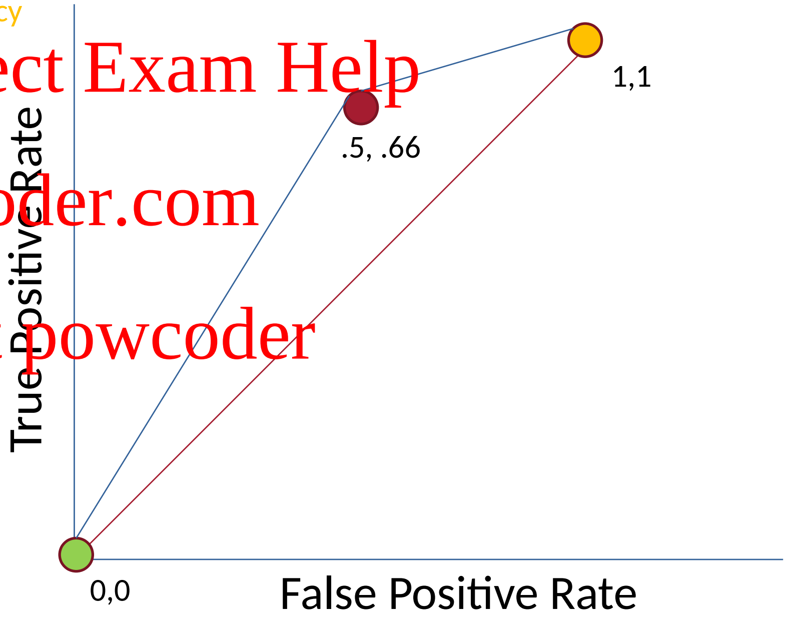
Not sensitive or specific

0.01 Cutoff

	Actual 1	Actual 0
Predicted 1	3	2
Predicted 0	0	0

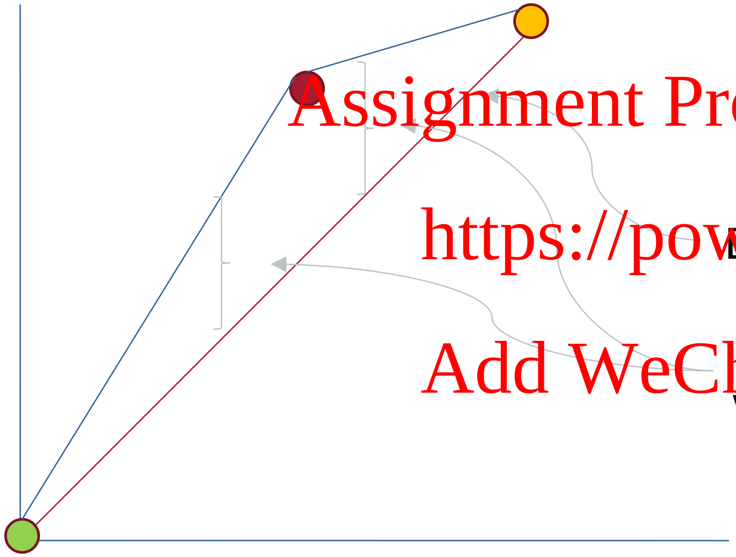
● True Positive Rate = 3/ 3
False Positive Rate = 2/2

Highly Sensitive not specific



*not proportional

Conceptually ROC & AUC



Diagonal Line: flipping a coin 50/50

Model "lift" better than random chance w/different cutoffs

In binary classification the AUC (area under the curve) is a KPI

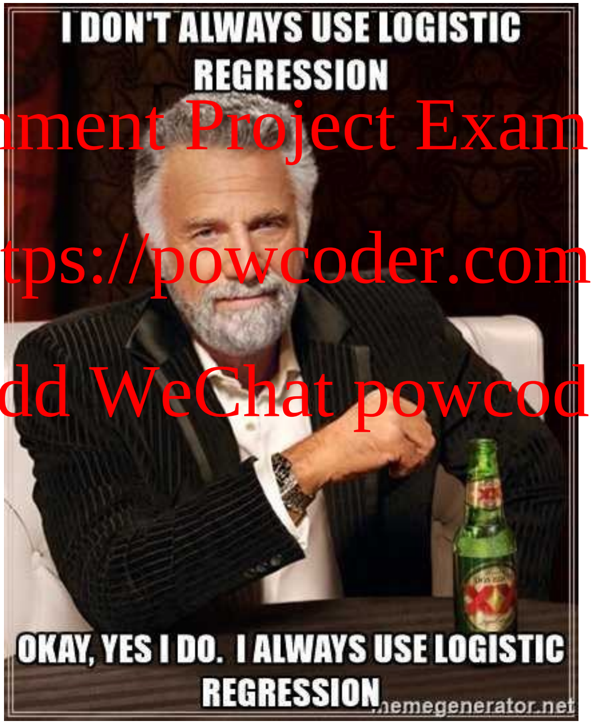
Logistic Regression Summary

- Logistic regression is similar to linear regression, except that it is used with a categorical response
- The predictors are related to the response Y via a nonlinear function called the *logit*
- As in linear regression, reducing predictors can be done via variable selection
- Logistic regression can be generalized to more than two classes

```
libray(nnet)
multiNomialLogit <- multinom(y ~ ., df)
```

Back to the script

Open B_fullyMarchMadnessREVISED.R


I DON'T ALWAYS USE LOGISTIC REGRESSION
OKAY, YES I DO. I ALWAYS USE LOGISTIC REGRESSION
memegenerator.net

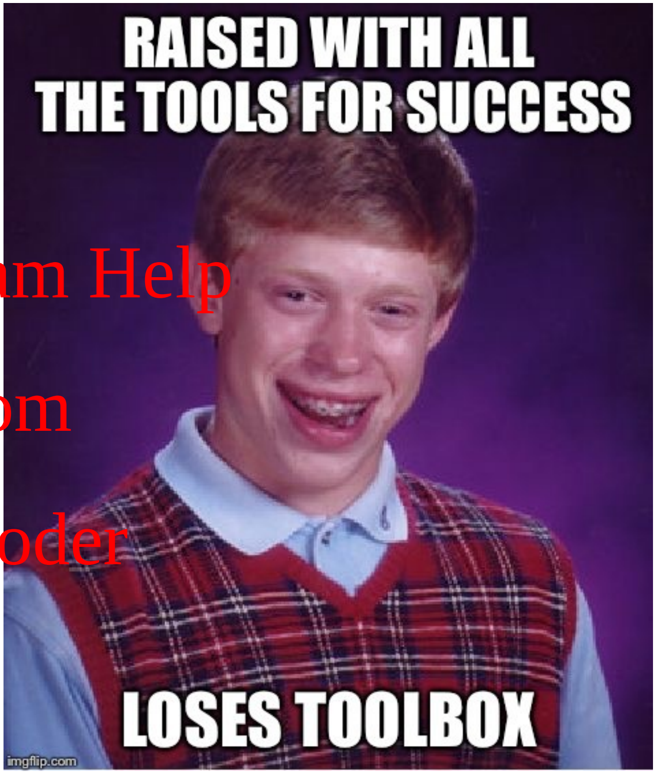
Your Data Mining Toolbox

<u>Previous Lessons</u>

- Some R Programming (R-studio) 🔧
- EDA (summaries, column and row exploration) 🔧
- Knowledge of Data Preparation (vtreat) 🔧
- Basic Visualization (plot, ggplot) 🔧
- Regression (continuous predictions) 🔧

<u>After today</u>

- Binary Classification (logistic regression) 🔧
- **More complex algorithms**



Regression & Logistic Regression are two good starting algorithms . Both put you on a path to more complex machine learning but more importantly you can start to frame business problems in terms algorithms can understand.

Agenda

Start	End	Item
		Logistic Regression
		Break
		East Side Vs West Side!
		Absenteeism KNN example

KNN

Now a new Classification Approach - KNN

Characteristics of KNN

- Data-driven, not heuristics (rules) based

- No parameters, beta coefficients, means time to predict or classify can be lengthy because each new record is scored against the existing training set.

- Makes no assumptions about the data e.g. outliers, non-normal distributions – all are accepted

Basic Idea

For a given record to be classified, identify nearby records

“Near” means records with similar predictor values X_1, X_2, … X_p

Classify the record as whatever the predominant class is among the nearby records (the “neighbors”)

Your brain makes similar associative leaps…the upcoming example proves it!

How to measure “nearby”?

The most popular distance measure is **Euclidean distance**

$$\sqrt{(x_1 - u_1)^2 + (x_2 - u_2)^2 + \cdots + (x_p - u_p)^2}$$

DON'T WORRY ABOUT THIS
R will handle it!

- Typically, predictor variables are first normalized (= standardized) to put them on comparable scales
- An easy and consistent method for normalization is to use `preProcess()` from `caret` but can also be done with `scale()`
- Without normalization, metrics with large scales dominate

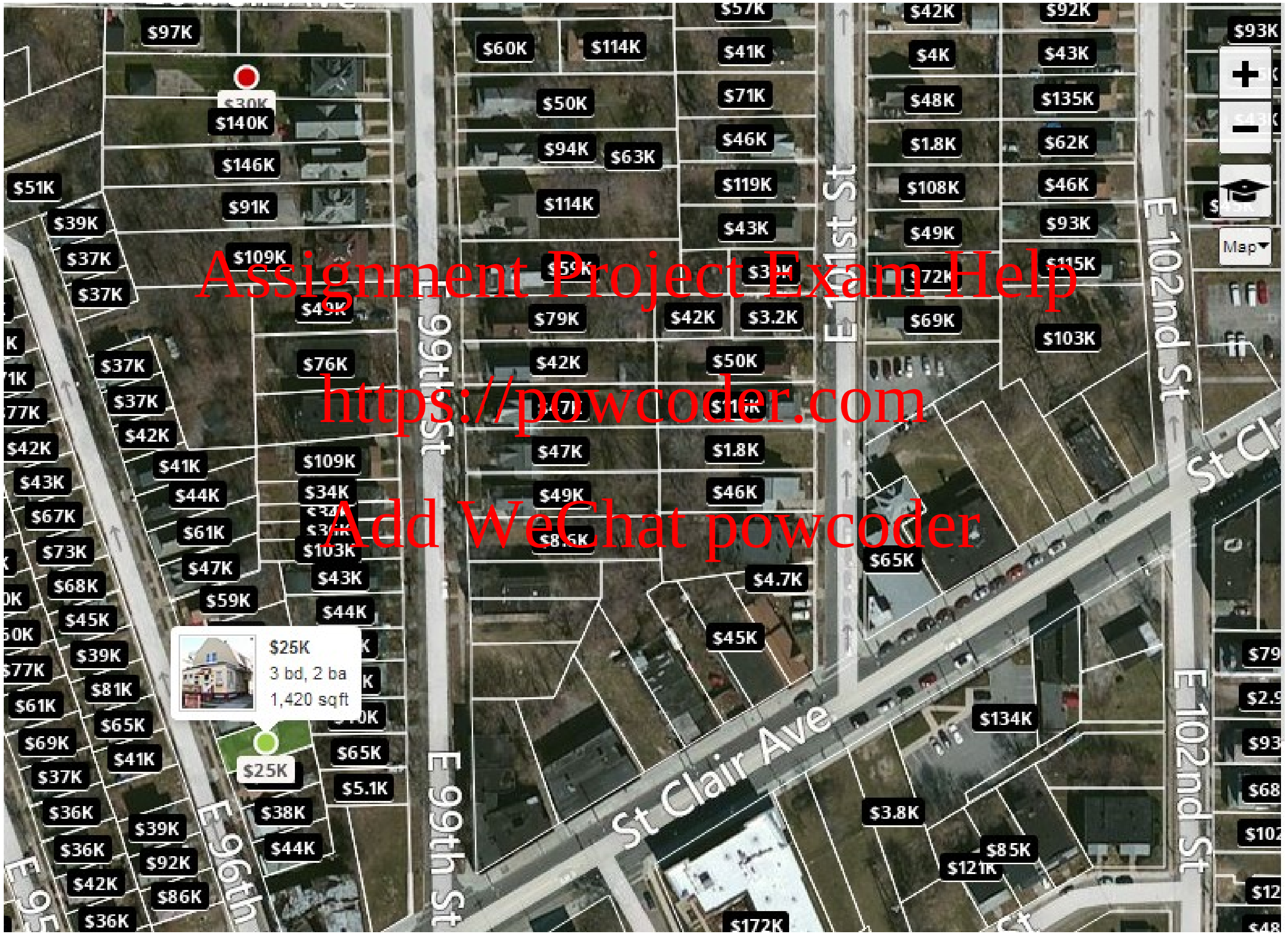
KNN Classification

Rep your HOOD!

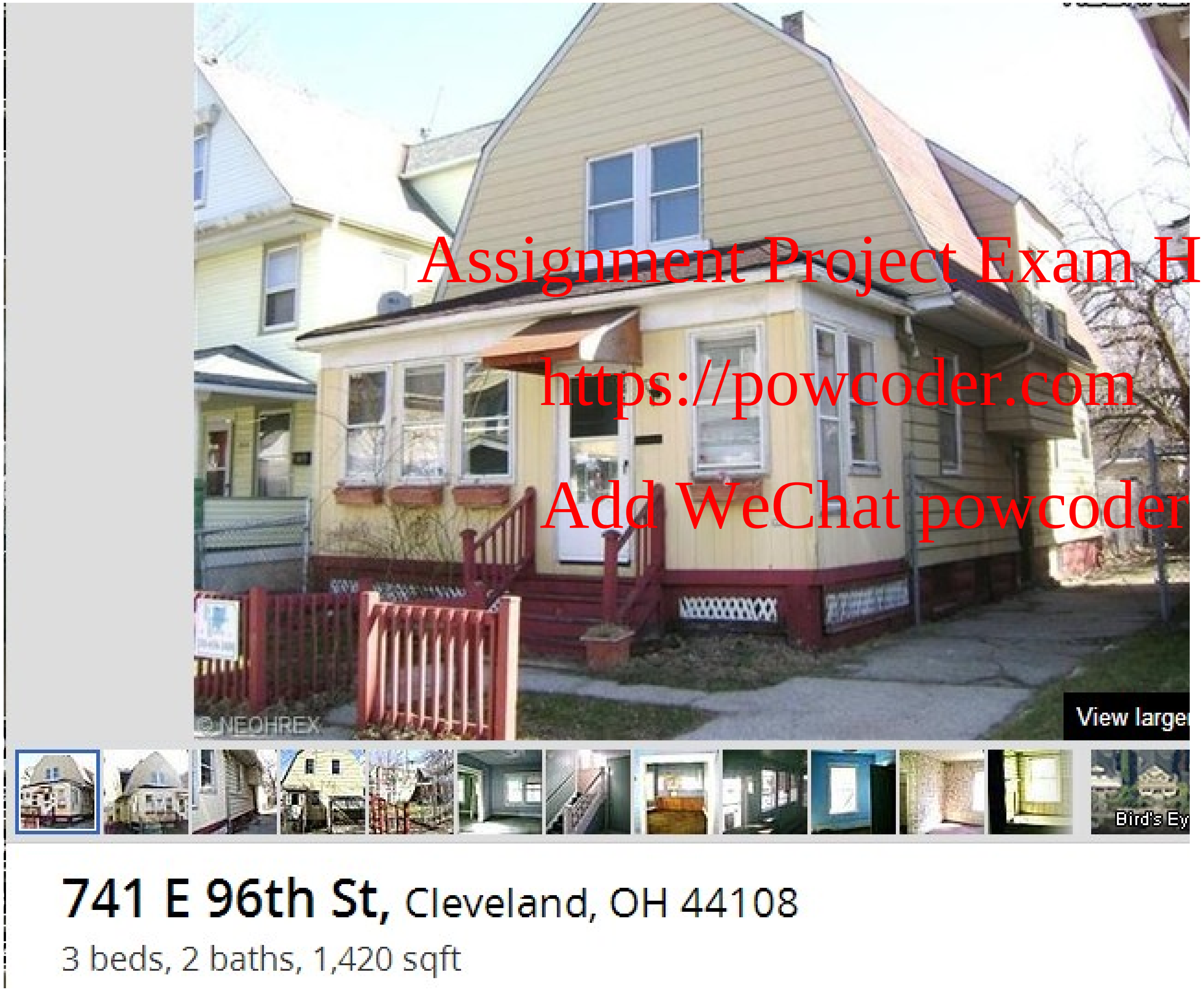
- From East 99th Street & St. Clair in Cleveland
- Talk about it a lot…

Lets try to understand if a house is in East or West Cleveland and Bone Thugs would live there based on some attributes.

Bone Thugs Hood on Zillow



Attributes of houses here…



741 E 96th St, Cleveland, OH 44108

3 beds, 2 baths, 1,420 sqft

- $25K
- 3 beds
- 2 baths
- 1,420 sqft
- 44108 zip

Collected a small data set comparing East Cleveland to West.

West Cleveland

Beds	Bath	SqFt	Price
4	4.5	4110	1.175M
5	1.75	1616	$155K
4	2	1480	$64K
4	4	2640	$279K
5	5	4175	$525K
5	2.5	1702	$120K
3	1	1582	$103K
3	2	1292	$100K
3	3	1780	$159K

East Cleveland

Beds	Bath	SqFt	Price
3	1	1181	$65K
3	1.5	1391	$39K
4	1	1424	$39K
4	2	1895	$30K
5	1	1607	$50K
4	1	1312	$11K
3	1	1152	$5K
4	1	1556	$81K
2	1	811	$46K

What patterns do we observe in this data?

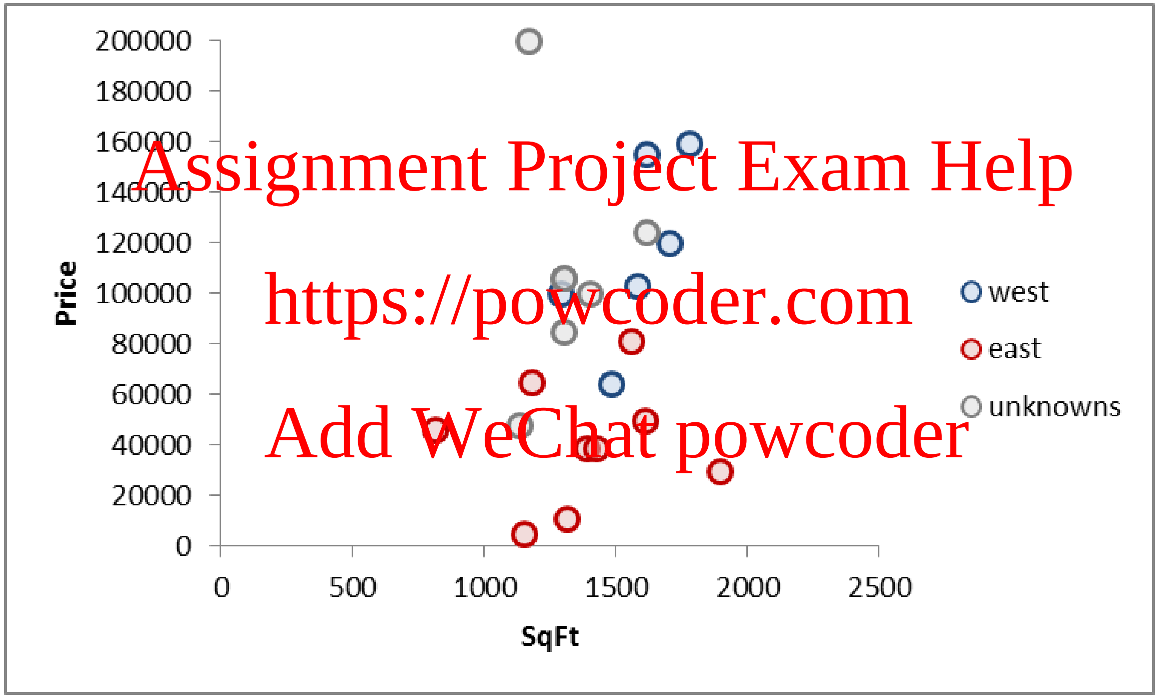
Here are some unknown houses…

East Side or West Side?

Beds	Bath	SqFt	Price
5	1.5	1136	$48K
6	3	4500	$259K
3	1.5	1300	$85K
3	2	1300	$106K
2	2	1170	$200K
5	2	2592	$95K
3	1	1398	$100K
3	2	1300	$106K
3	1.5	1614	$124K

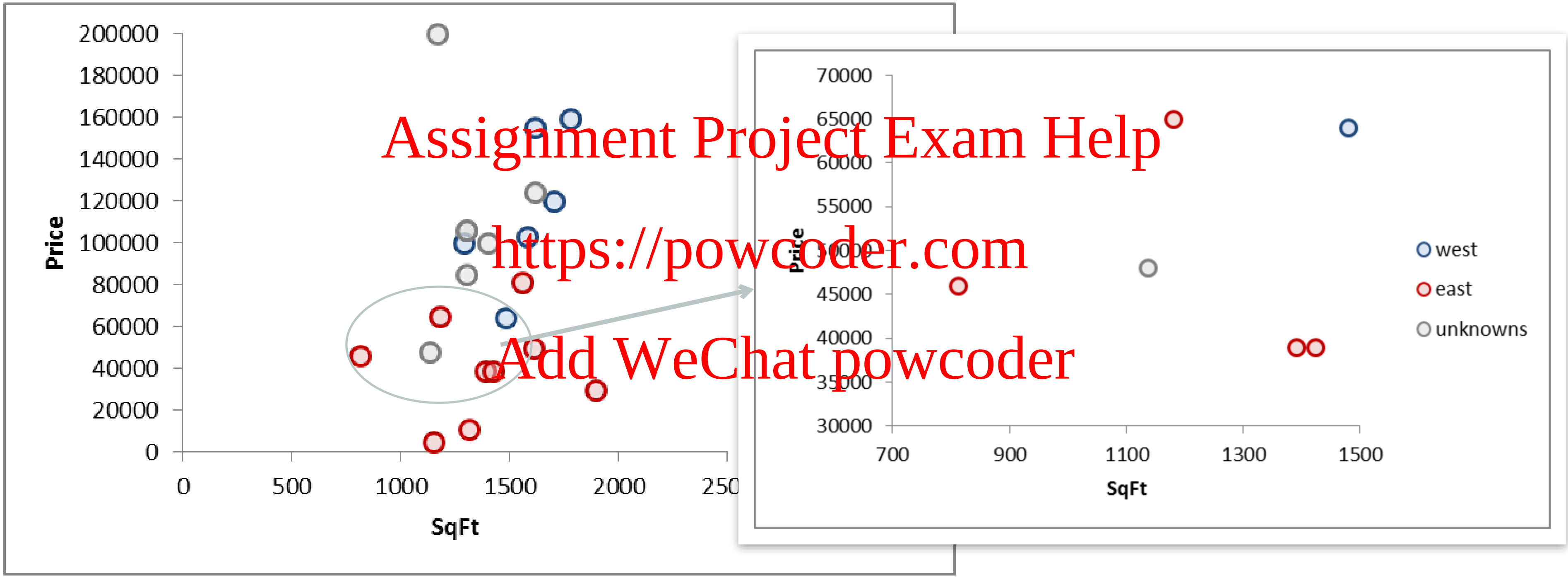
Let’s pick two house attributes, sqft and price

Plot East, West and Unknown houses on a scatter



KNN Measures the Euclidean distance between points

Lets zoom in to a specific point

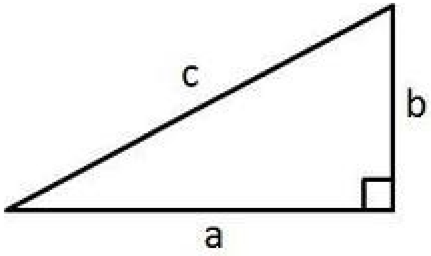


Without knowing distances or making calculations:
What side of the city do you think the unknown is?

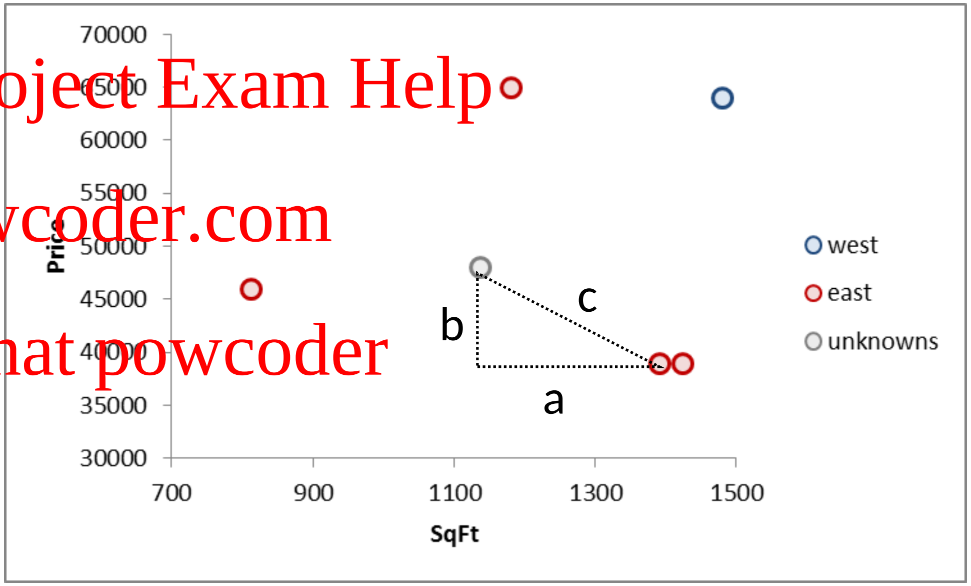
Euclidean Distance measures distance like a ruler

Remember Pythagorean Theorem?

- $A^2+B^2=C^2$

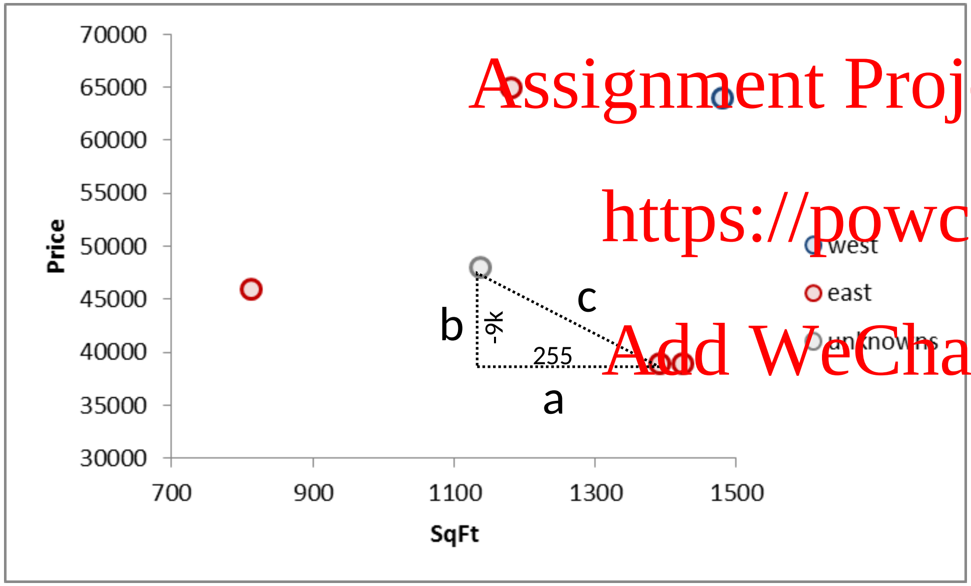


In our example…



Euclidean Distance measures distance like a ruler

In our example…



Define the segment values

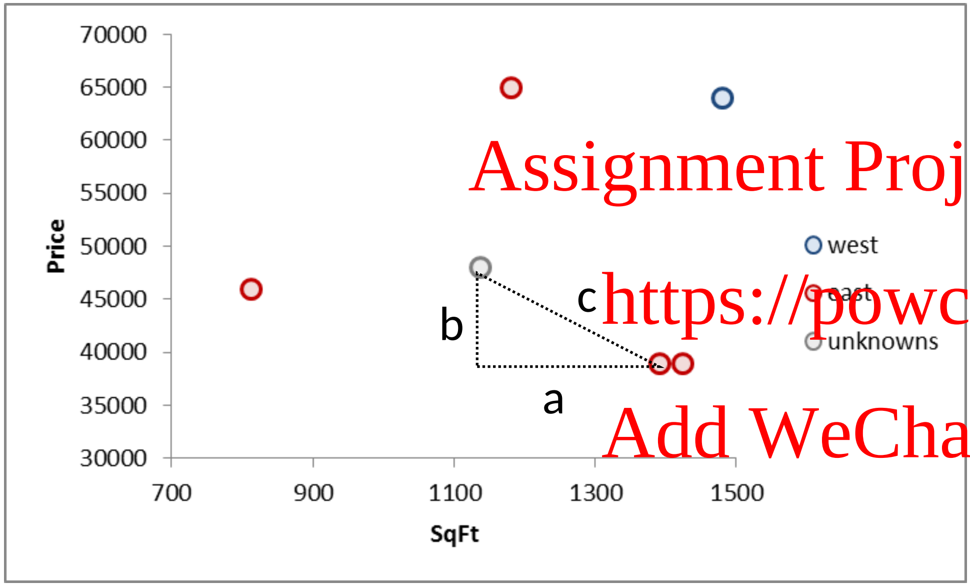
EastSidePt = (1391sqft, $39K)
UnknownPt = (1136sqft,$48K)

aSegment = 1391-1136 = 255sqft

bSegment = $39k-$48k = $-9k

Euclidean Distance measures distance like a ruler

In our example…



Math time…

EastSidePt = (1391sqft, $39K)

UnknownPt = (1136sqft,$48K)

aSegment = 1391-1136 = 255sqft

bSegment = $39k-$48k = $-9k

$A^2 + B^2 = C^2$

$\$-9000^2 + 255^2 = c^2$

$\$81,000,000 + 65025 = c^2$

$\$81,065,025 = c^2$

$\text{Sqrt}(\$81,065,025) = c$

9003 = c

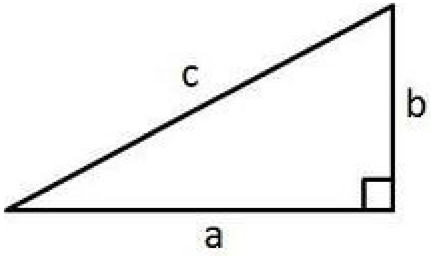
The distance between the unknown and the closest East side point is 9003*.

We didn’t normalize (put all attributes on the same scale) so you can see that large impact price has on the distance moving from 9000 to 9003 but that’s not the point…this is just to show you a distance calc.

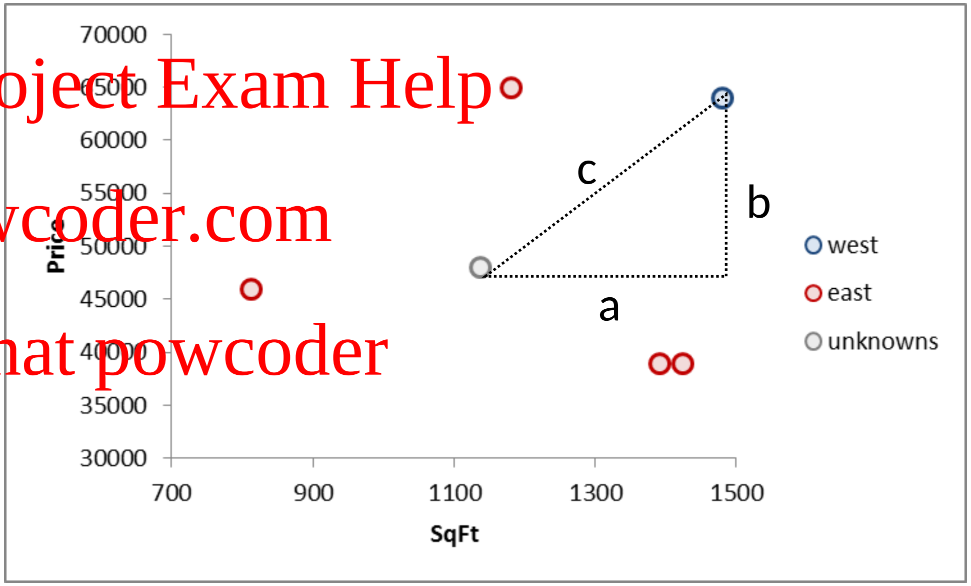
Euclidean Distance measures distance like a ruler

Remember Pythagorean Theorem?

- $A^2+B^2=C^2$



In our example…

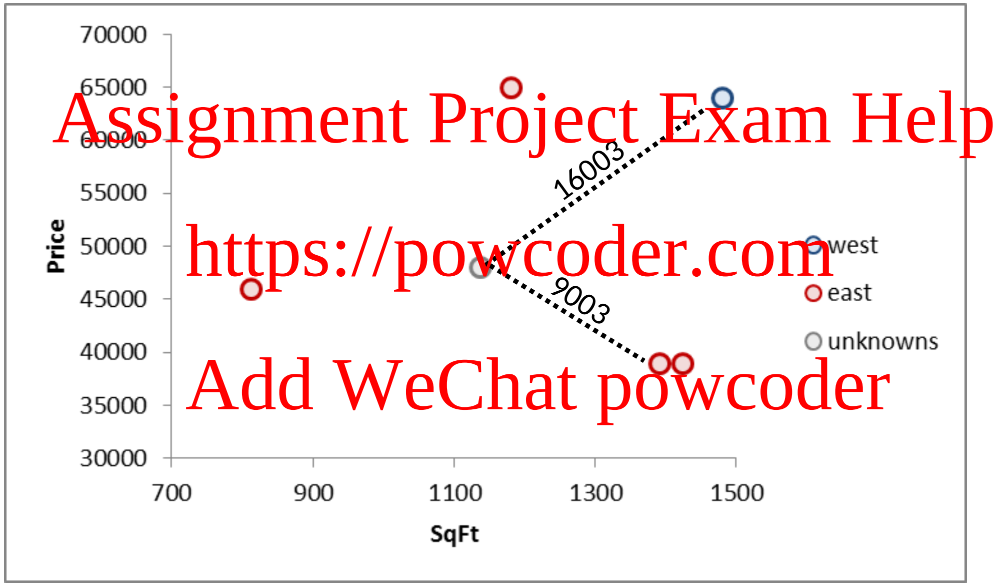


West Side (1480sqft, \$64K)
Unknown (1136sqft, \$48K)

Differences (45sqft, \$17K)
b a

∴ Distance
or C = 16003

Your guess K=1



With K = 1, the single nearest neighbor, what is the class? What about k=2? K=3?

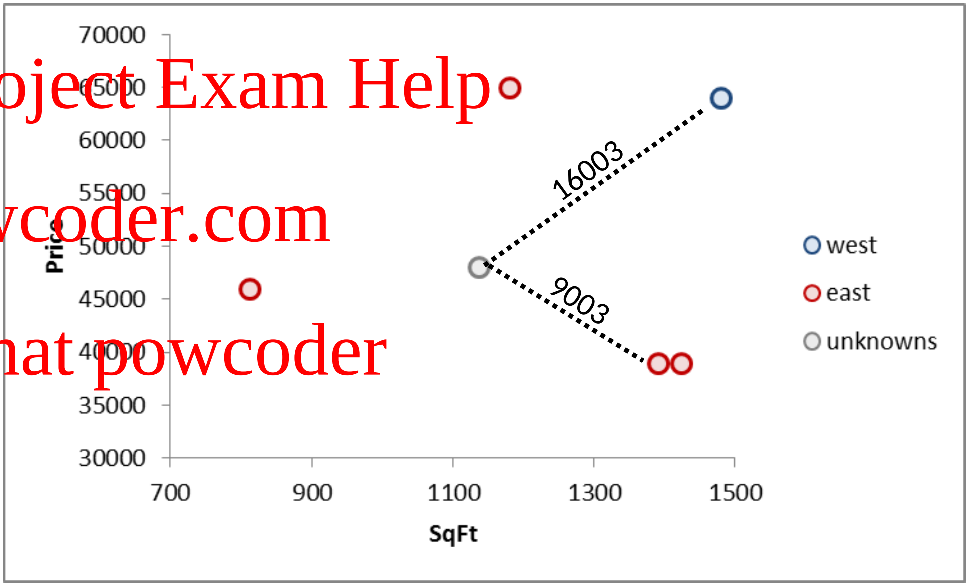
K = 1

This unknown case is in…

- East Cleveland

  - 5 Beds
  - 1.5 Bath
  - 1136 sqft
  - $48K

In our example…



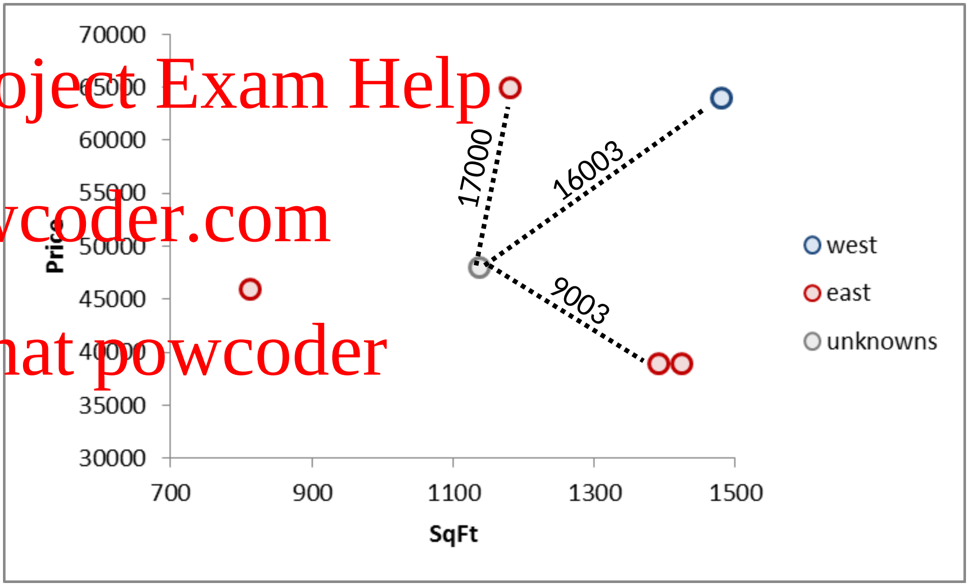
The KNN algorithm would have correctly identified East Cleveland if K = 1. Remember the algorithm performs this along more than 2 dimensions, in hyperspace.

K is a tuning parameter the practitioner chooses.

You will have to specify how many neighbors are to be looked at.

- Measured in hyperspace (many attributes not just 2)
- Ties are randomly chosen for even number K but can be avoided using odd number K.
- Returned results can be either the class (east or west) or the probability of a particular class.

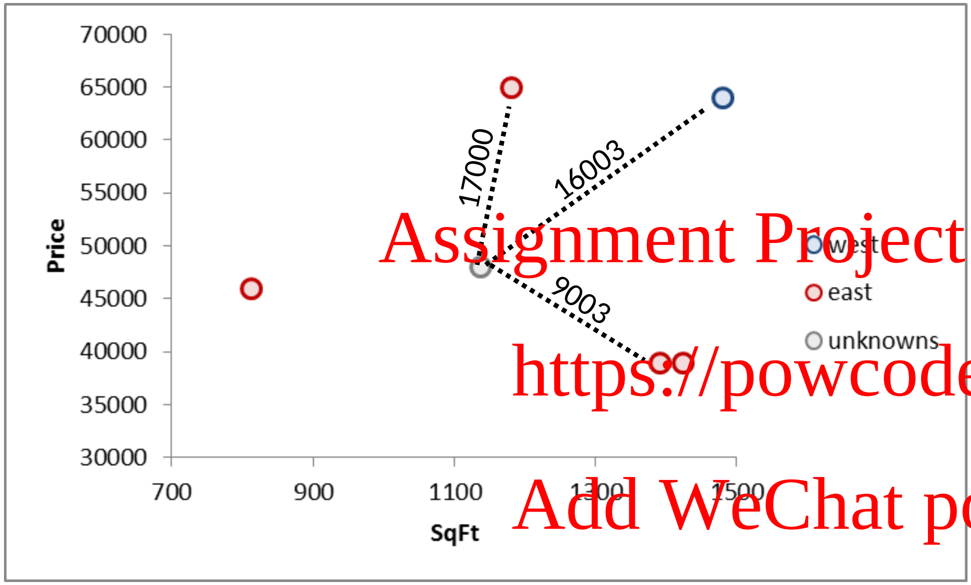
In our example…



*scale is misleading because of attributes order of magnitude
Remember to normalize!

K=1 of only these 2 variables would say it is East Cleveland with a 100% certainty.
K=3 of only these 2 variables would say it is East Cleveland with a 66% probability.
Review confusion table to get to an acceptable K.

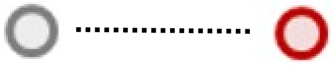
K is a tuning parameter the practitioner chooses.



*scale is misleading because of attributes order of magnitude
Remember to normalize!

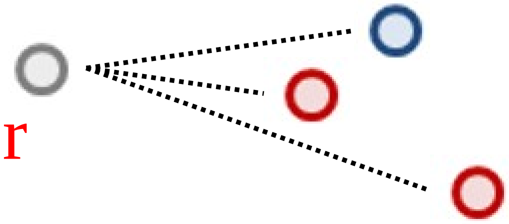
Nearest Neighbor

- K = 1



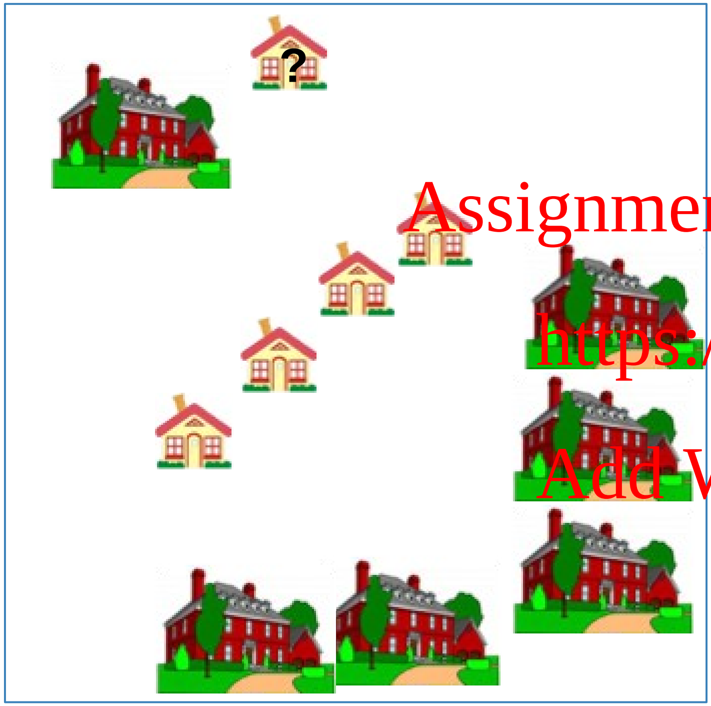
1 Red Neighbor / k =1 = 100% red

- K = 3



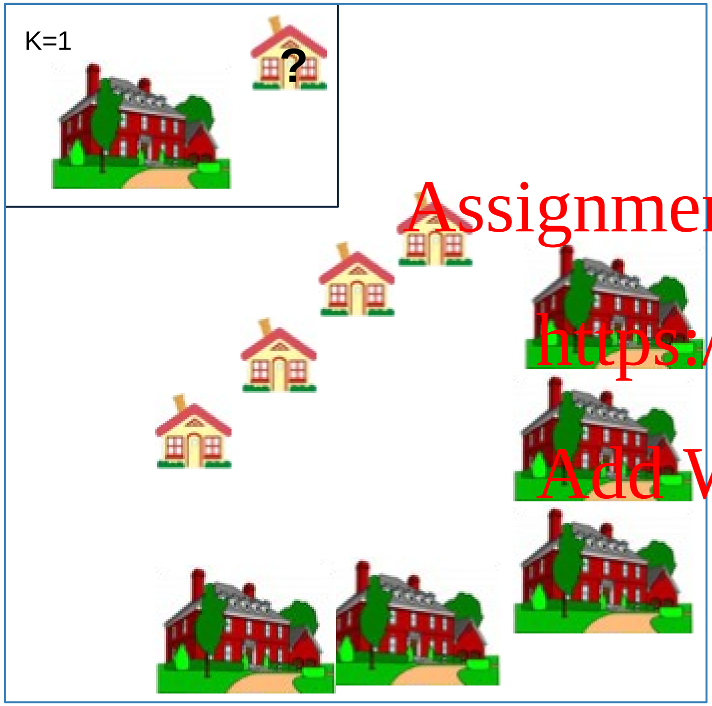
2 Red Neighbor / k =3 = 66% red

Special K!



Perhaps more so than other methods, tuning your KNN is of the utmost importance.

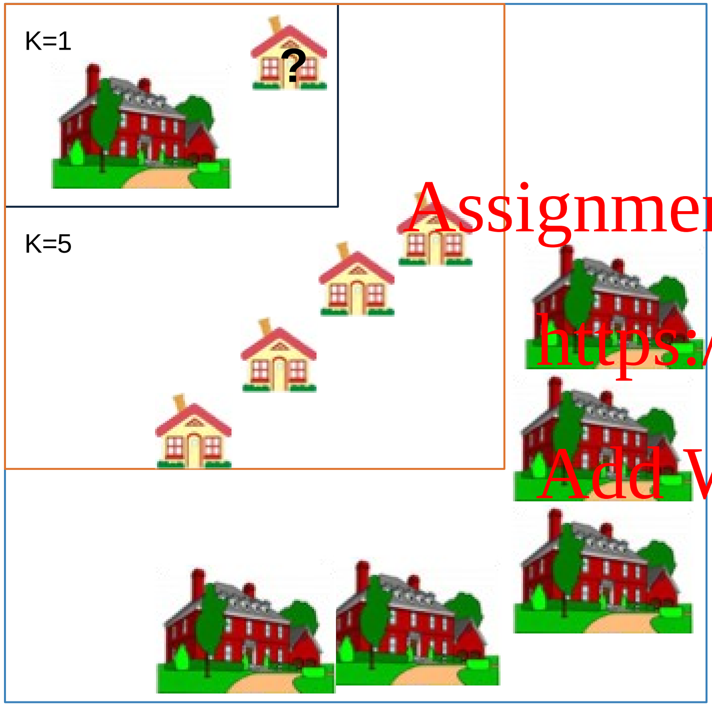
Special K!



K	Big House	Lil House	Guess	Actual	Notes
1	1	0	Big House	✗	Local Structure

Perhaps more so than other methods, tuning your KNN is of the utmost importance.

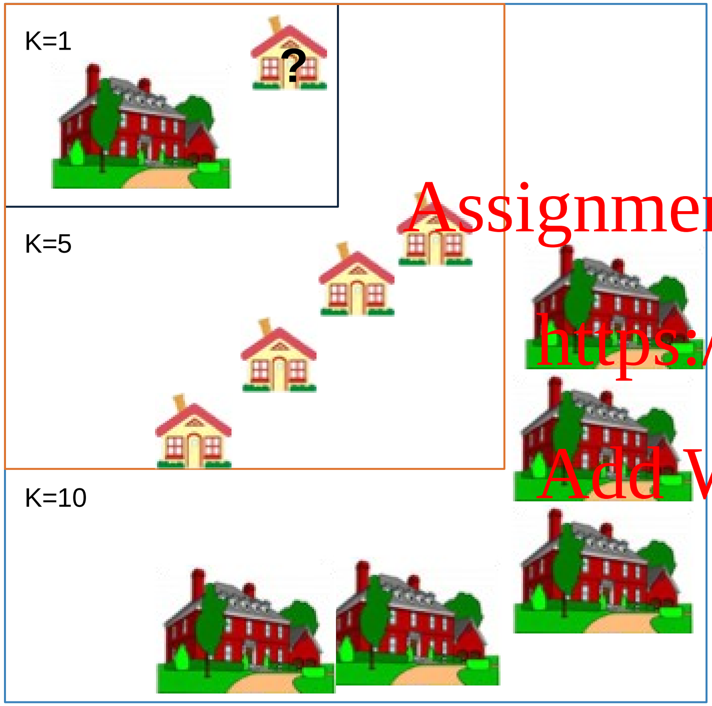
Special K!



K	Big House	Lil House	Guess	Actual	Notes
1	1	0	Big House	✗	Local Structure
5	1	4	Lil House	✓	Just right

Perhaps more so than other methods, tuning your KNN is of the utmost importance.

Special K!



K	Big House	Lil House	Guess	Actual	Notes
1	1	0	Big House	✗	Local Structure
5	1	4	Lil House	✓	Just right
10	6	4	Big House	✗	Majority Structure

Perhaps more so than other methods, tuning your KNN is of the utmost importance.

Choosing k

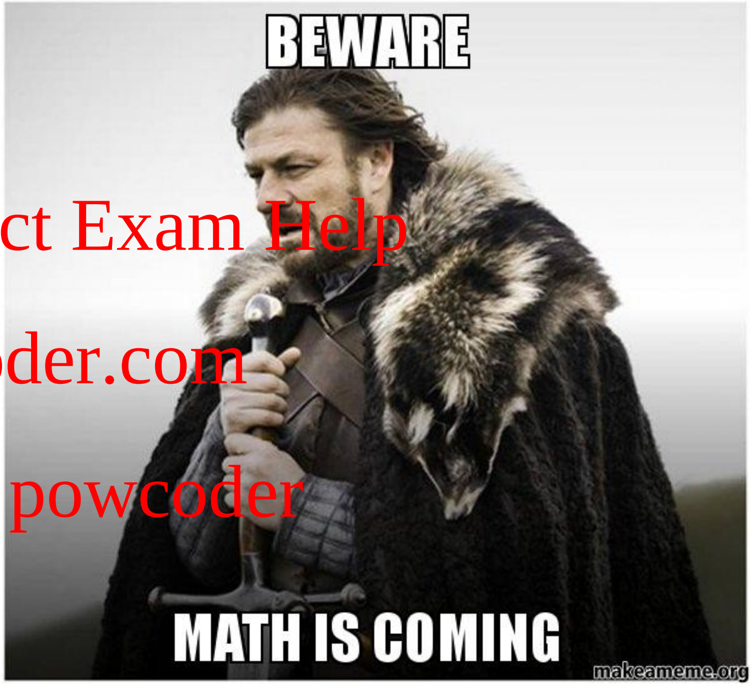
- *K* is the number of nearby neighbors to be used to classify the new record
  - *K*=1 means use the single nearest record
  - *K*=5 means use the 5 nearest records all have a “vote”
- Typically choose that value of *k* which has lowest error rate in validation data
- Use odd numbers to avoid ties

Let’s predict the probability of “Class A”

K=	Possible Outcomes
1	Nearest neighbor is class A so 100% therefore class B probability is 0%
2	Nearest 2 neighbors agree class A so 100%, or all neighbors are class B so 0% or neighbors are split so outcome is 50%
3	Nearest 3 neighbors agree class A so 100%, or all neighbors are class B so 0%, or they split 33% or 66%. With a cutoff of 50% you can still make a classification.
4	Nearest 4 neighbors agree class A so 100%, all neighbors are class B so 0%, or they split 25%, 50% or 75%. Cases of 50% probability are troublesome to determine same as K = 2.

Open C_normalization example_REVISED.R

VTREAT will NOT scale your data!!



This script will show you how R “scales” and “centers” data to be on the same magnitude. We did it as part of the preprocessing lesson.

Agenda

Start	End	Item
		Logistic Regression
		Break
		East Side Vs West Side!
		Absenteeism KNN example

Classifying Absenteeism at Work

Business Context:

Workers that are absent are costly to businesses.

- In the US absenteeism is estimated to cost $225.8B or $1685 per employee EACH YEAR*
- Understanding absenteeism reasons could lead employers to offer new benefits (like in office medical services) to reduce absenteeism

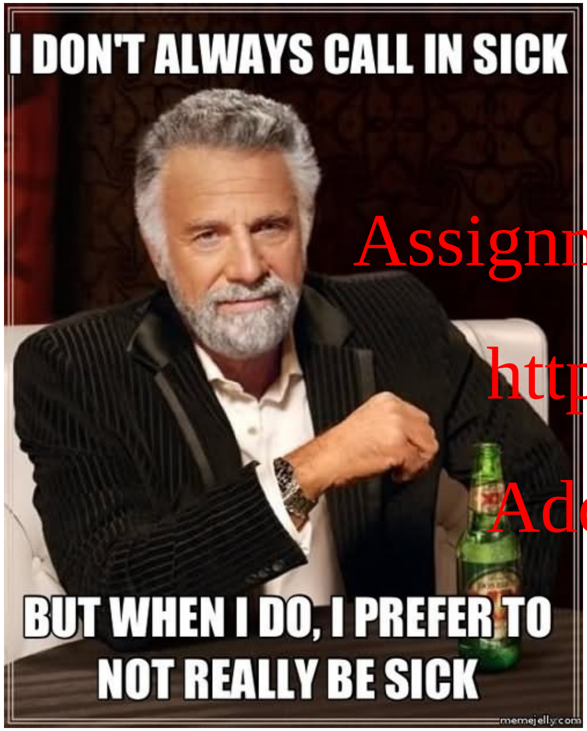
Dataset Source and Info:

- http://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work
- The database was created with records of absenteeism at work from July 2007 to July 2010 at a courier company in Brazil.
- 740 Rows * 21 Attributes

Our business objective is to classify the reason the employee is missing so we can possibly tailor employee support. "What is the probability the absent employee is out because of "dental consultation" or "medical consolation"?

https://www.cdcfoundation.org/pr/2015/worker-illness-and-injury-costs-us-employers-225-billion-annually

Open D_knn_example_classification.R


I DON'T ALWAYS CALL IN SICK
BUT WHEN I DO, I PREFER TO NOT REALLY BE SICK
memejelly.com

When someone calls in the operator lists a reason for the absence.

```
Reason.for.absence
lowFreq   :272
reason_0 : 43
reason_13: 55
reason_19: 40
```

We will build a KNN model classifying the reasons employees have been absent. A model like this can help classify new absent employees so we can learn what to offer to mitigate absenteeism costs and not ask employees directly.

Using K-NN for Prediction (Continuous)

- Instead of "majority vote determines class" use average of response values

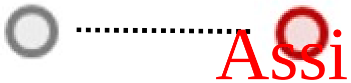
- May be a weighted average, weight decreasing with distance

KNN has drawbacks but can be used for both prediction and classification so it demonstrates flexibility in that regard.

KNN is helpful for both business problems.

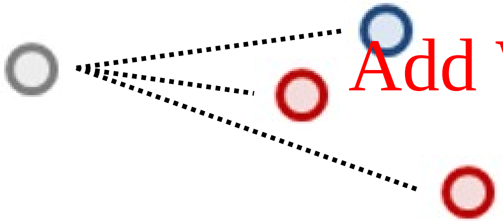
Class & Probability

- K = 1



1 Red Neighbor / k =1 = 100% red

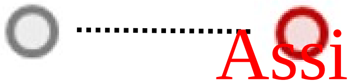
- K = 3



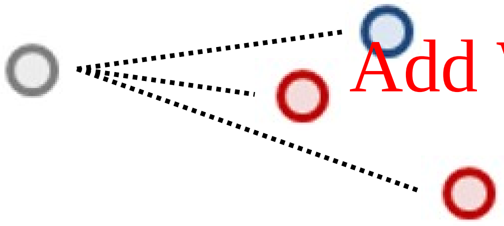
2 Red Neighbor / k =3 = 66% red

KNN is helpful for both business problems.

Classification

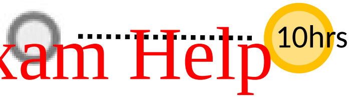
- K = 1 

1 Red Neighbor / k =1 = 100% red

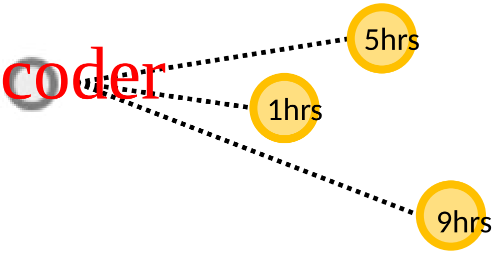
- K = 3 

2 Red Neighbor / k =3 = 66% red

Prediction

- K = 1 

Average of k =1 neighbor = 10

- K = 3 

Average of k =3 neighbors
= (9+5+1) / 3 = 5hrs

Open E_knn_example_prediction.R

Historical human resource records show the amount of time called out.

```
Absenteeism.time.in.hours
Min:0
1st Qu.: 2
Median: 3
Mean:   6.92
3rd Qu.: 8
Max: 120
```

Now let’s predict how much time an absent employee will miss based on their attributes. The predicted outcome is now `Absenteeism.time.in.hours` which ranges from 0 to 120hrs.

KNN Summary

- Simple concept, useful for classification & prediction
  - Classification – majority class of nearest neighbors wins
  - Prediction – average value among nearest neighbors

- Find distance between known and unknown records

- “Curse of dimensionality” – need to limit # of predictors

- Slow to predict new records –
  - *non-parametric* i.e. for every new record it must measure the distance to all data points in the training set, so the model object has to also keep all of the original data. In contrast a parametric model like linear regression has to only keep the **beta coefficients** so its much faster.

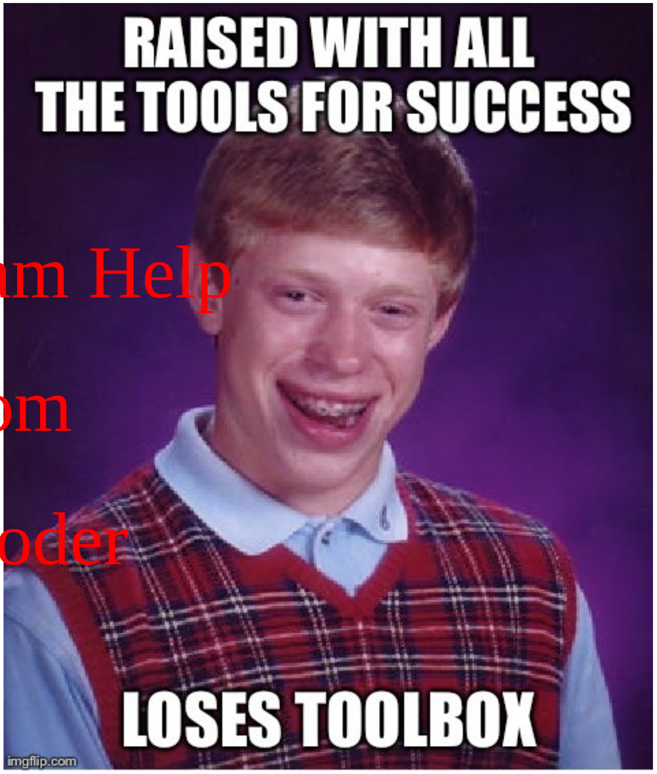
Your Data Mining Toolbox

Previous Lessons

- Some R Programming (R-studio)🔧
- EDA (summaries, column and row exploration)🔧
- Knowledge of Data Preparation (vtreat)🔧
- Basic Visualization (plot, ggplot) 🔧
- Regression (continuous predictions)🔧

Today

- Binary Classification (logistic regression)🔧
- KNN (continuous & classification – binary or multi)🔧



The KNN algo is a real machine learning algorithm which can solve binary classification, multi-classification & continuous problems!

Appreciation for your hard work…

- Homework Help on 7.2:
  - z_homework_supplemental_studentVersion.R

- Read Chapter 9