Decision Trees & Random Forests

Review steps

SAMPLE

1. Load data
   - read.csv()
2. Partition to avoid overfitting
   - sample(), df[idx,] etc.

EXPLORE

3. Exploratory Data Analysis
   - summary(), plot(), table() etc.

MODIFY

4. Prepare data for modeling
   - Vtreat:designTreatmentsC/N(), prepare()

MODEL

5. Model
   - Regression, Logistic Regression, KNN etc.
6. Get Results (classification or prediction)
   - predict()

ANALYZE

7. Key Performance Indicators
   - RMSE, Accuracy, MAPE

Agenda

Start	End	Item
		Decision Trees - explanation
		Decision Tree Example
		A to Z Decision Trees scripting example
		Break
		Random Forests

Decision Trees

- By observing the data and splitting it into sections, rules are created for either prediction or classification problems.

- Mimics a subject matter expert…pre data mining days.
  - Before data mining, an experienced marketing bank manager may have said "let's call our *married* customers *over* 25 that have at *least a college education* to see if they want another loan."

Key Ideas

Recursive partitioning: Repeatedly split the records into two sections so as to achieve maximum homogeneity of outcome within each new section

Pruning the tree: Simplify the tree by pruning peripheral branches to avoid overfitting – measure and reduce complexity

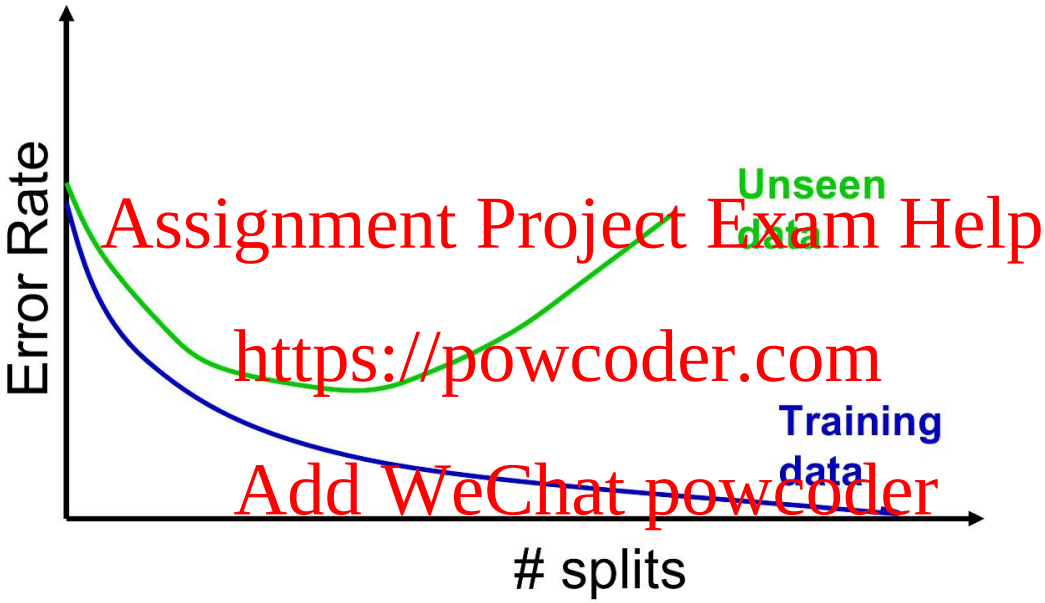
Advantages of Decision Trees

- Understandable, rules are human readable; executives love looking at them
- Light weight, fast
- Easy to implement…logic can be built in Excel even
- Variable selection is automatic
- No assumptions of data
- Works with minimal data preprocessing

Disadvantages of Decision Trees

- Overfitting!! You could create rules down to individual records so you get perfect accuracy (100% purity in each section). This wouldn't generalize to new unseen data.

Disadvantages of Decision Trees

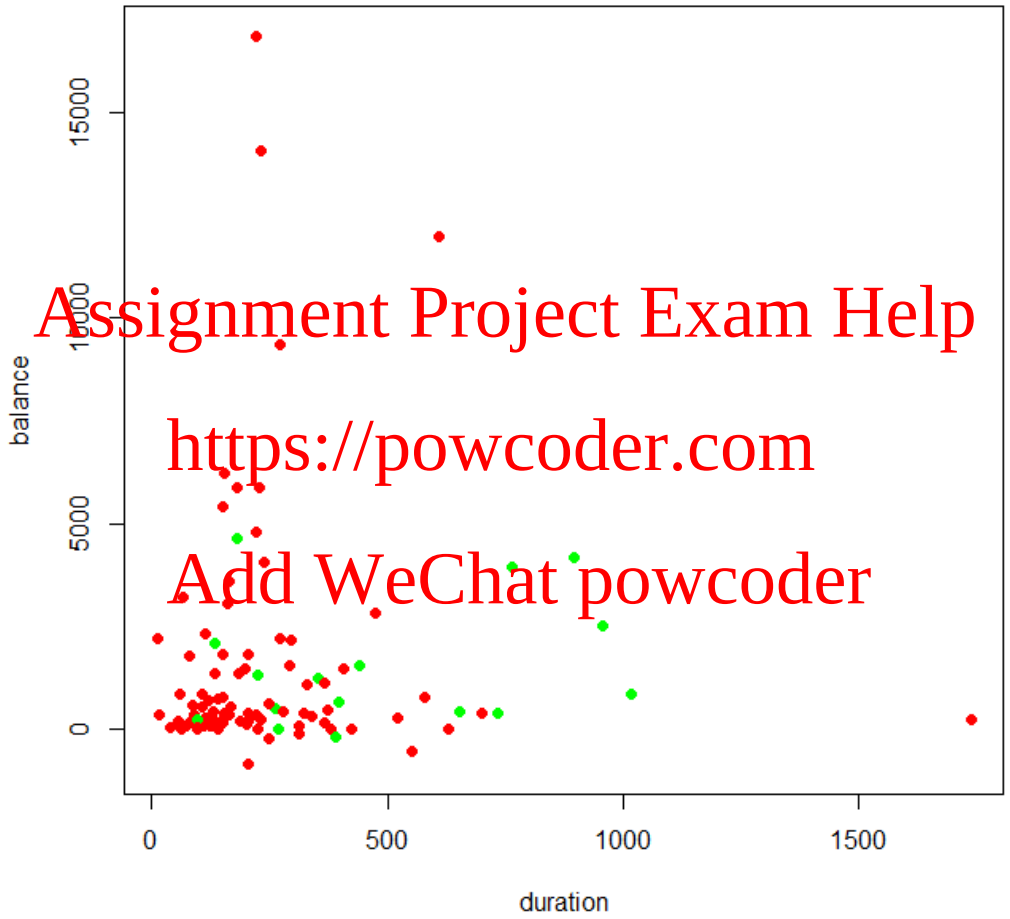


This is why having a training, test and holdout partition is important when making a decision tree. In production it is also important to review results of a model periodically to ensure the historical patterns aren't evolving.

Recursive Partitioning Steps

- Pick one of the predictor variables, x_i
- Pick a value of x_i, say s_i, that divides the training data into two (not necessarily equal) portions
- Measure how "pure" or homogeneous each of the resulting portions is
  "Pure" = containing records of mostly one class (or, for prediction, records with similar outcome values)
- Algorithm tries different values of x_i and s_i to maximize purity in initial split
- After you get a "maximum purity" split, repeat the process for a second split (on any variable), and so on

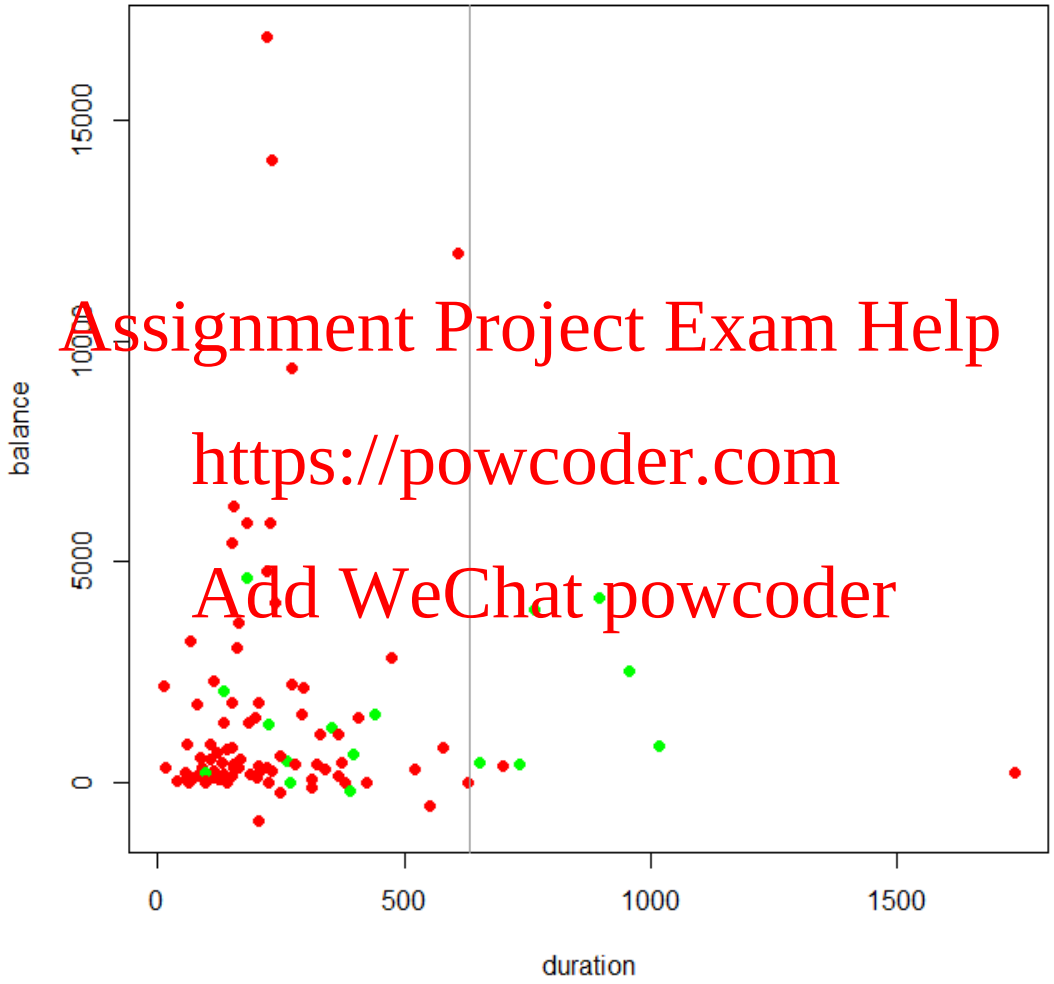
Suppose this is our data



Let's classify offer acceptance among real bank customers using a decision tree. Green means they accepted the offer to open a new deposit account.

https://archive.ics.uci.edu/ml/datasets/bank+marketing

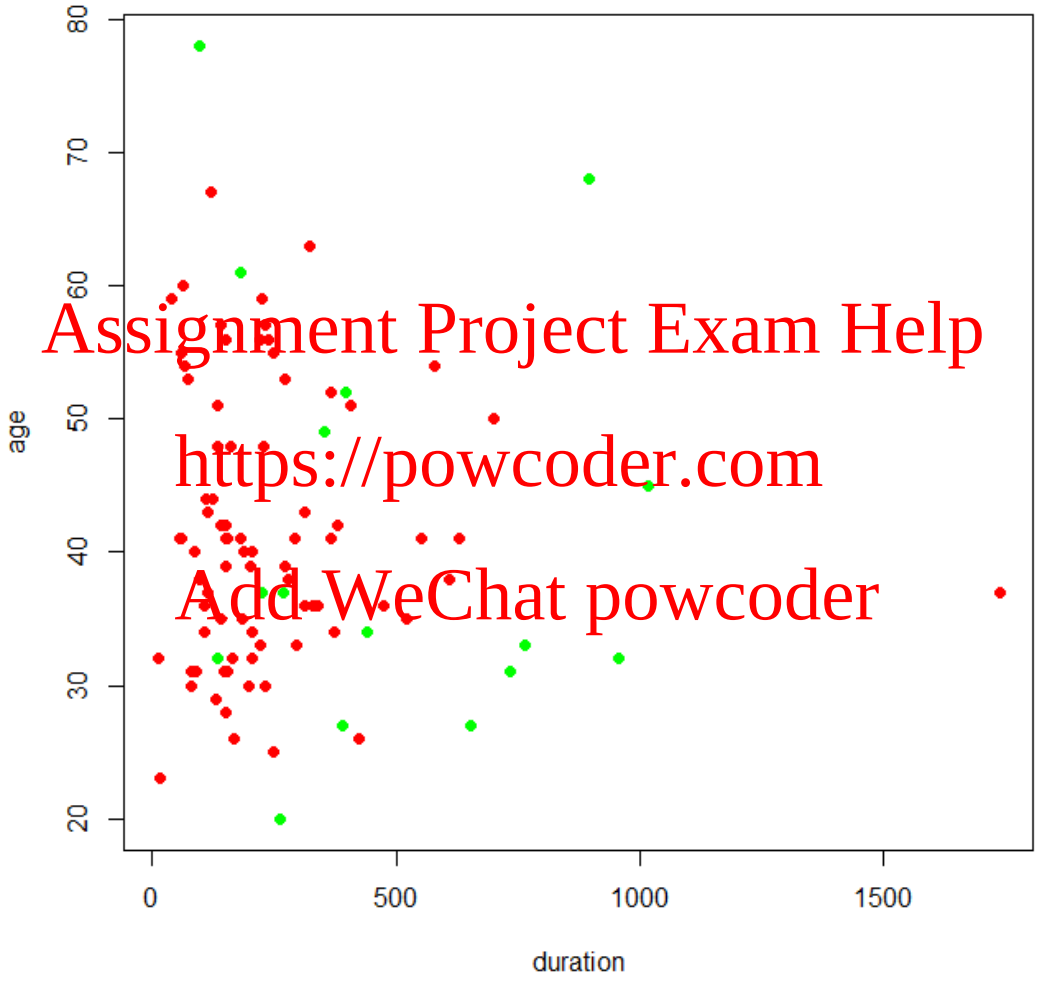
Suppose this is our data



Purity = 6 accepted among 8 offers

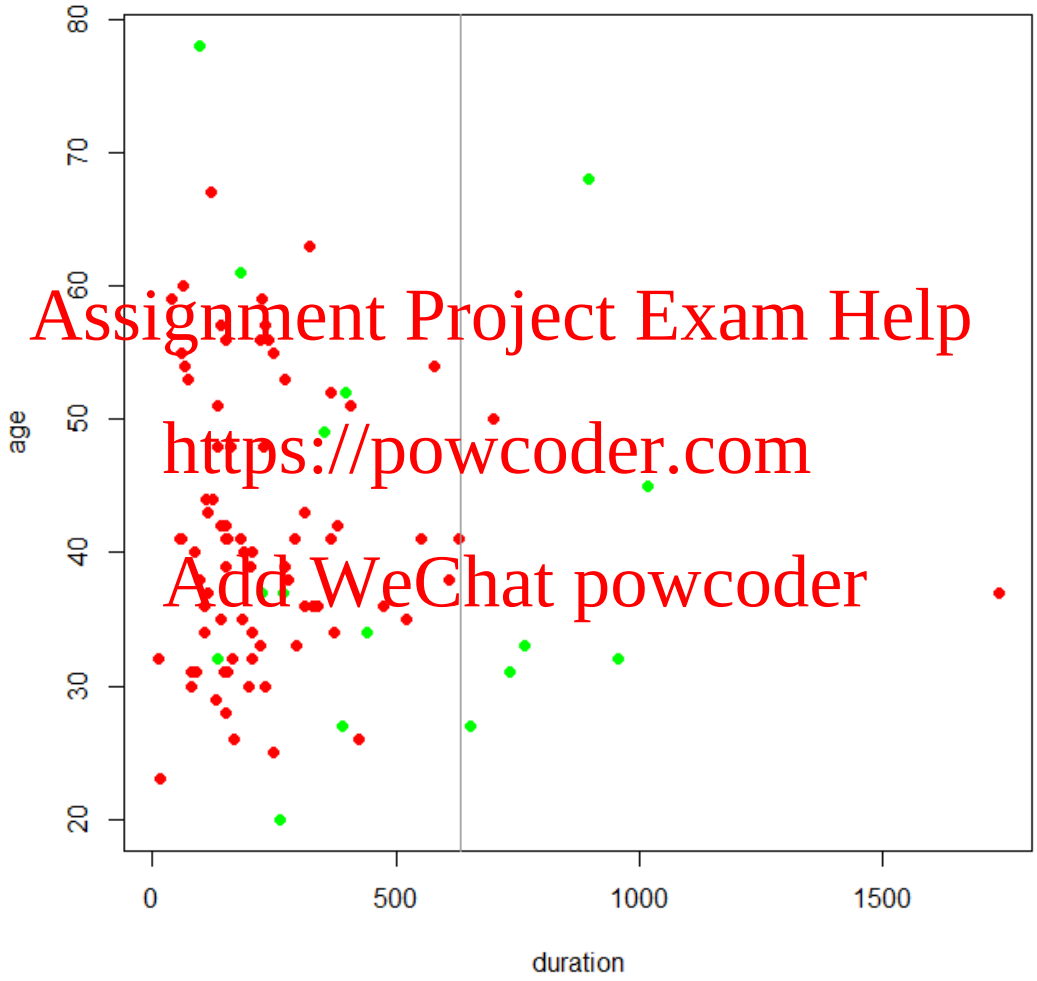
There are a lot of accepted offers with account holders of duration >635 days

Now another view of the data



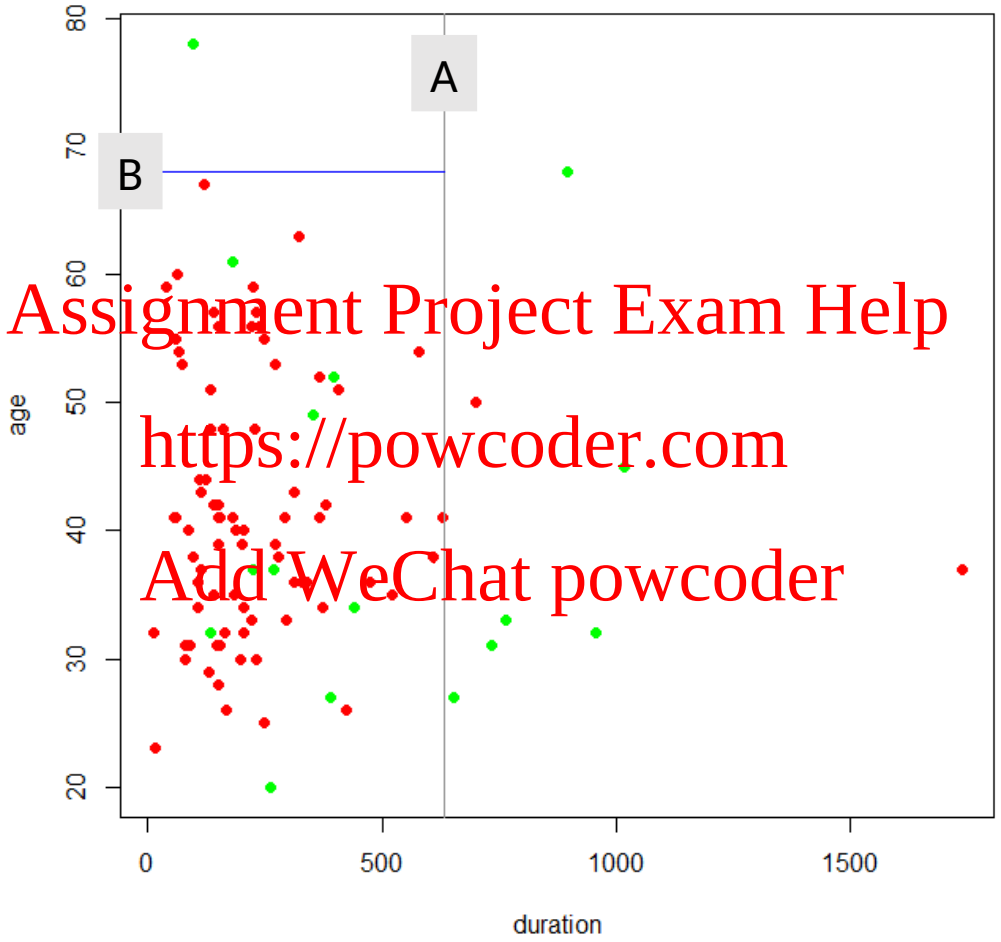
Here we see another dimension, age of the account holder along with duration.

With the rule duration > 635



The first rule still holds true but now we can think about adding new rule layers.

New Rule



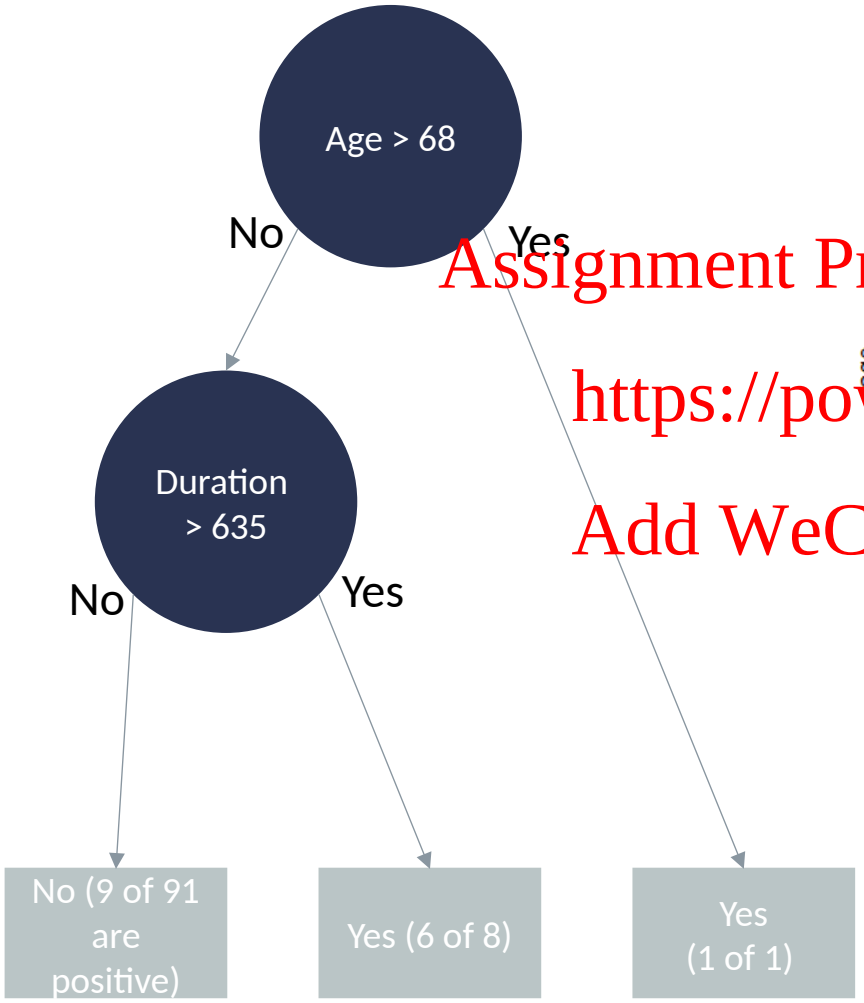
- Purity of split "A"= 6 accepted among 8 offers
- Purity of split "B" = 1 of 1

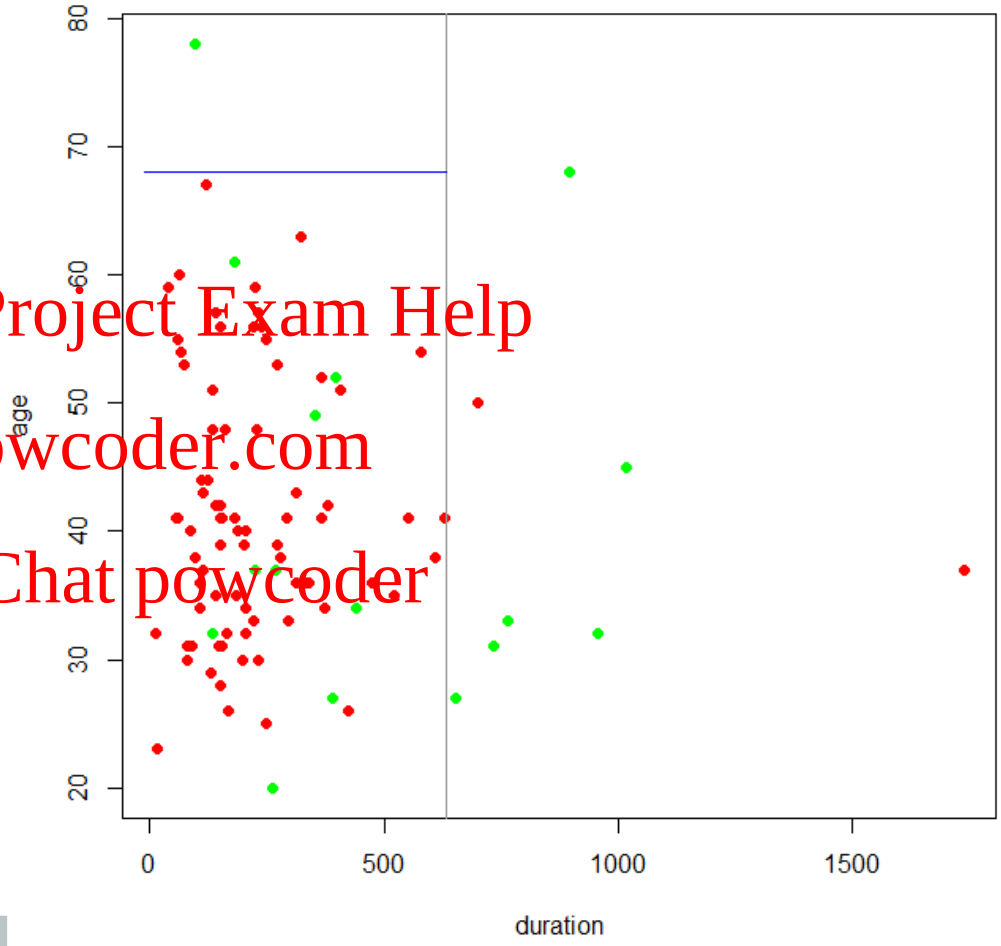
In this toy example, we can add another rule stating age > 68 to capture another positive response and create another rule.

Our rule set is like Plinko!

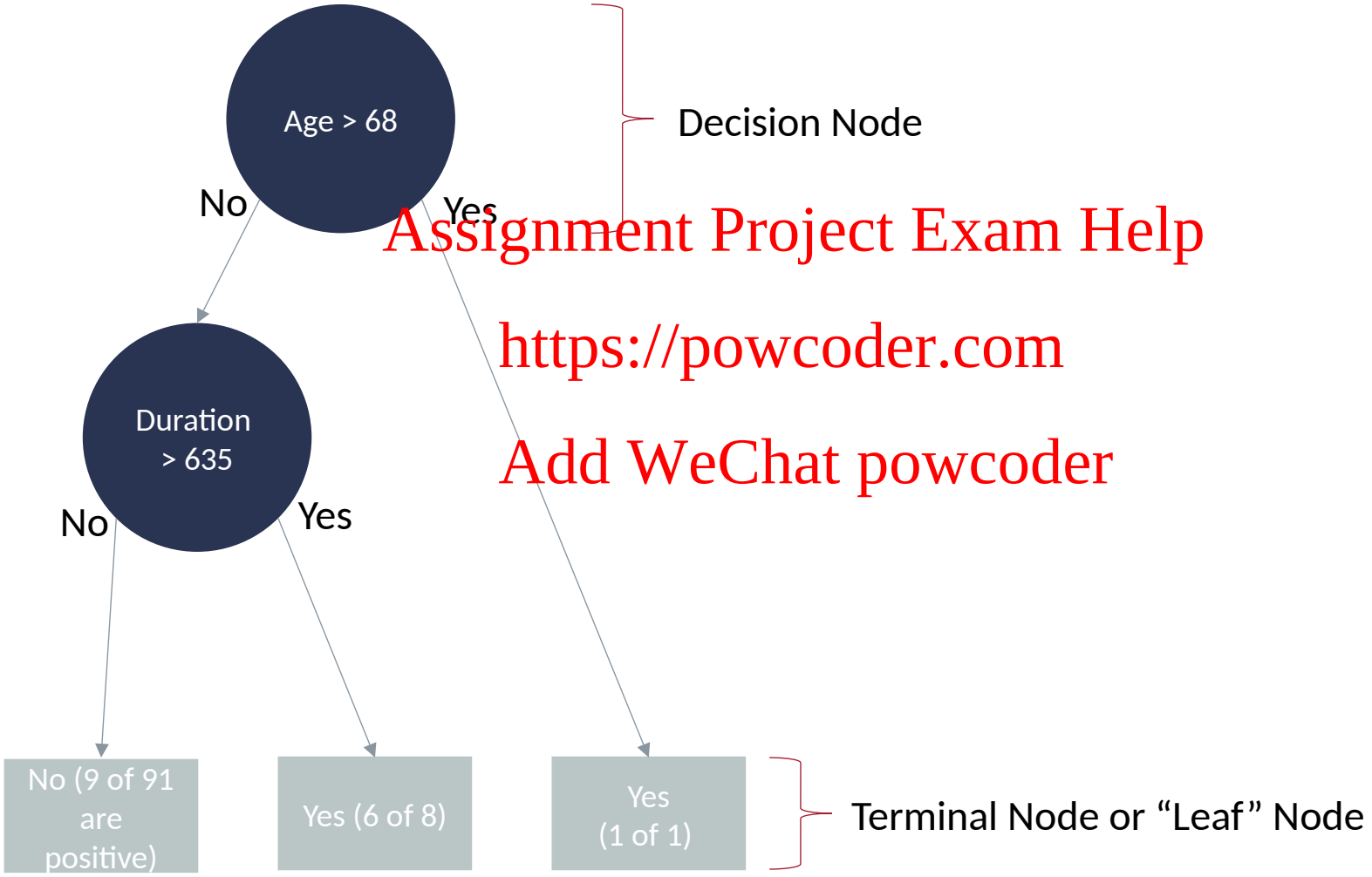
With our rule tree we can score a new potential customer to call.

2 Rule Tree

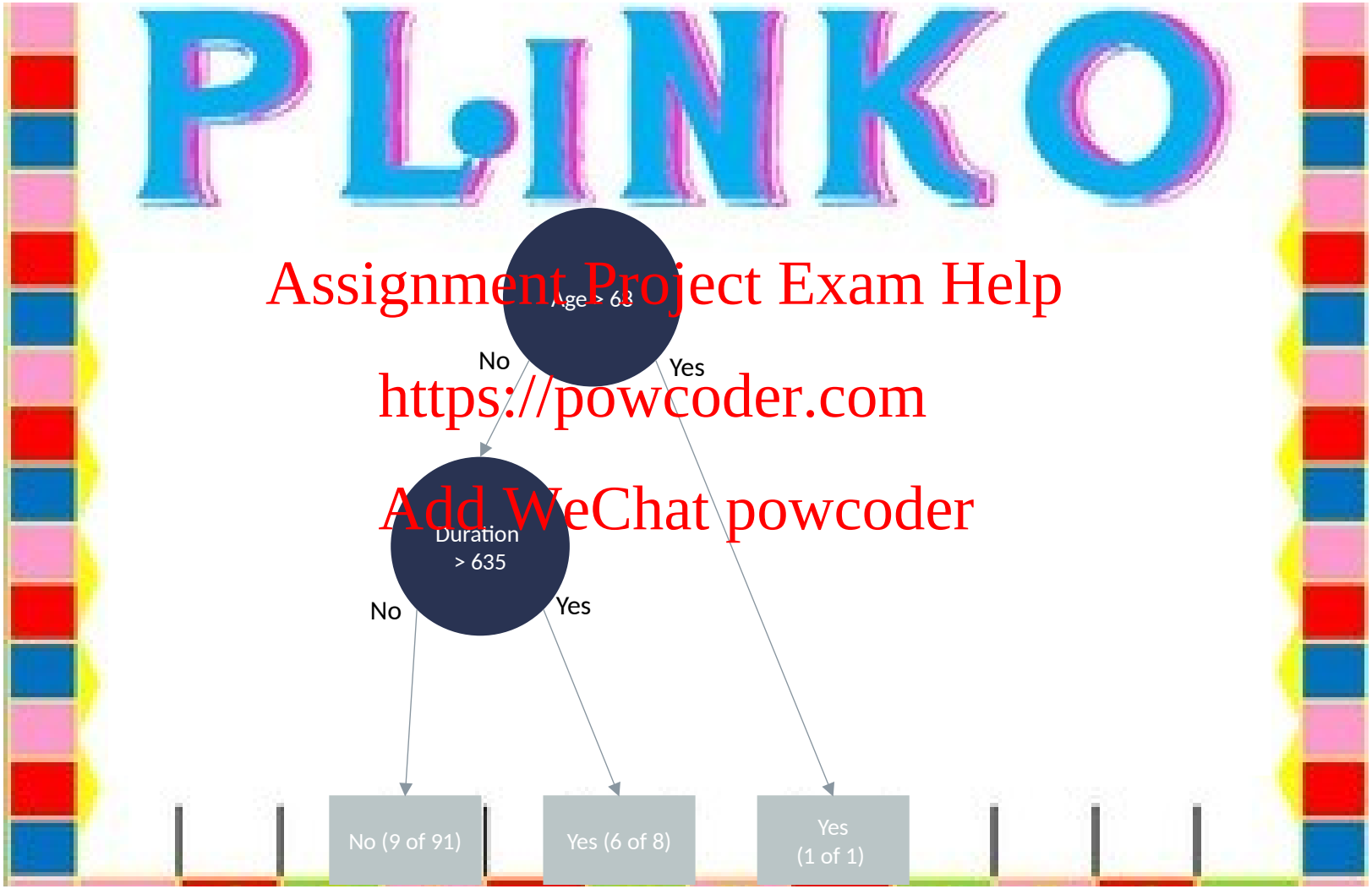




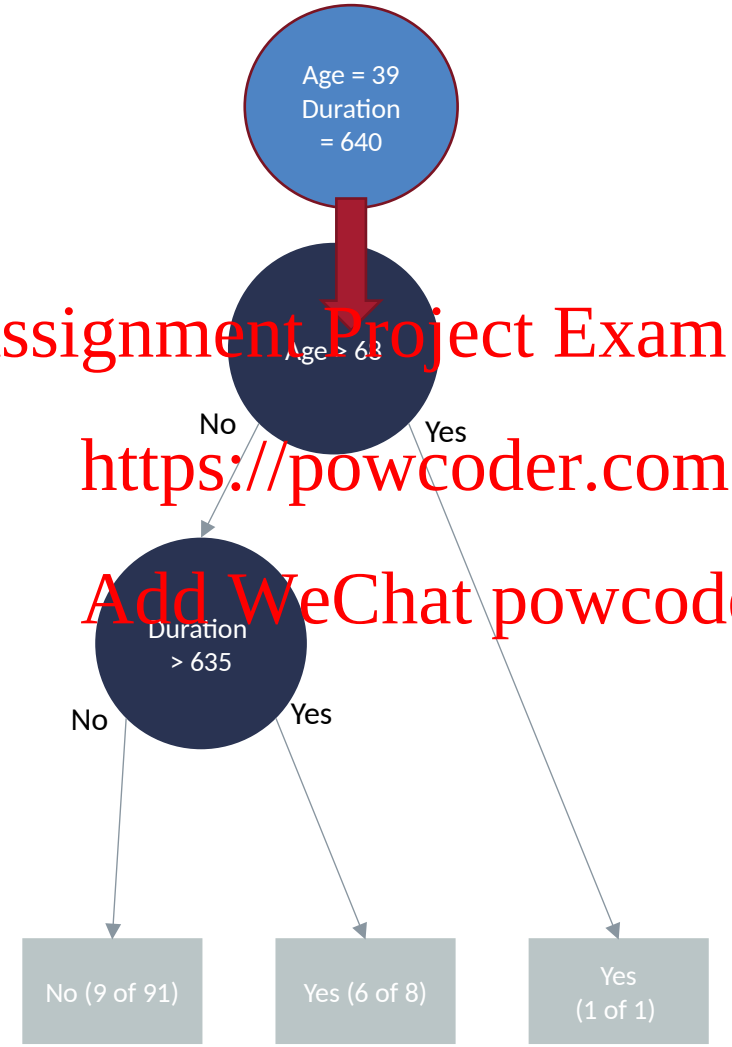
2 Rule Tree



Let's drop a new record down our plinko tree

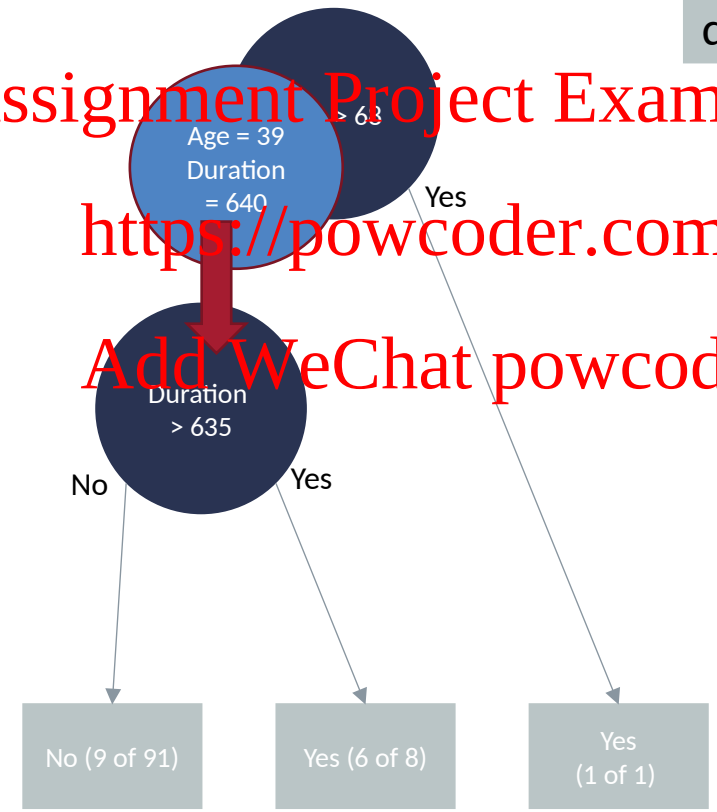


Let’s drop a new record down our plinko tree



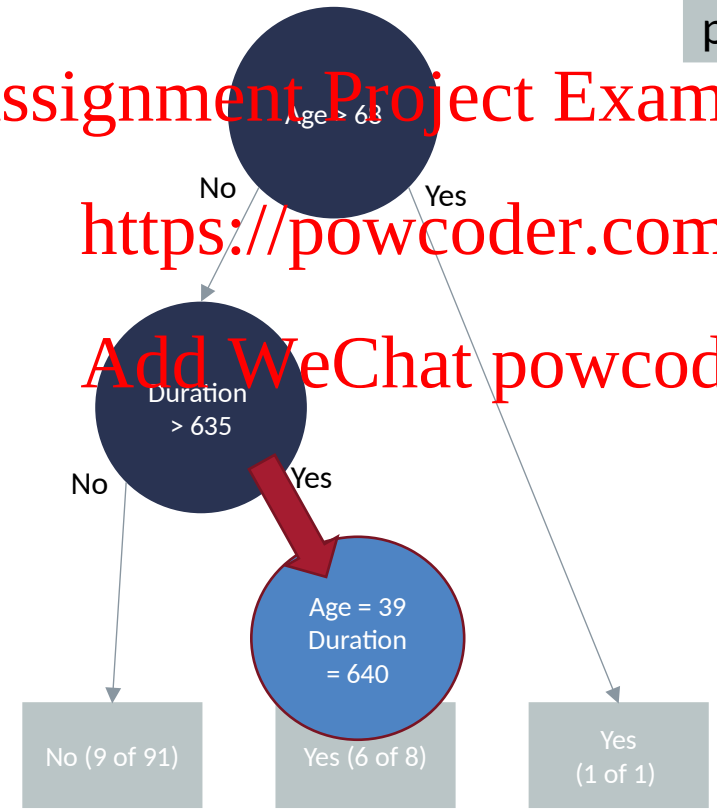
Let's drop a new record down our plinko tree

The age was less than 68 so the record drops to the next decision point.

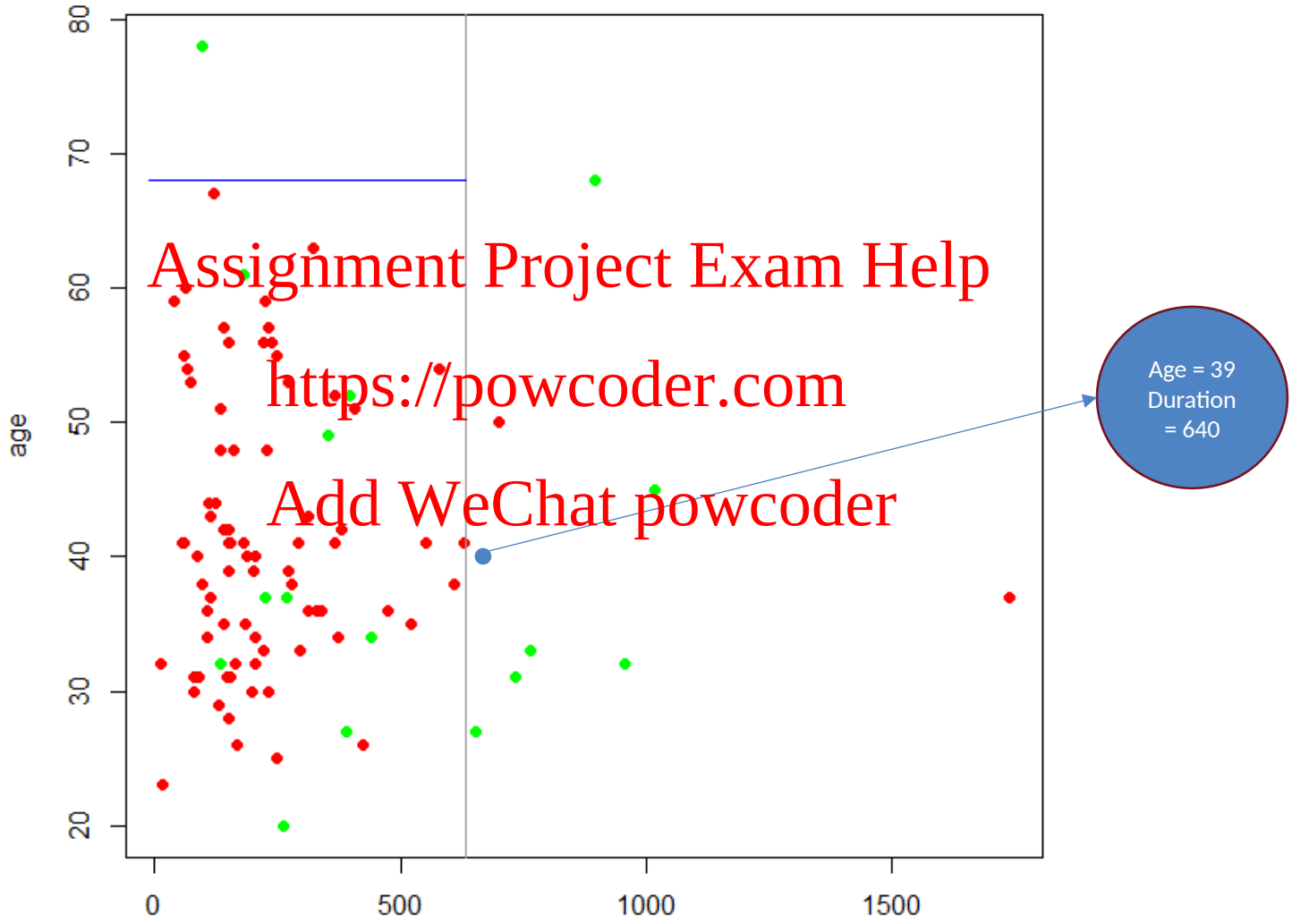


Let’s drop a new record down our plinko tree

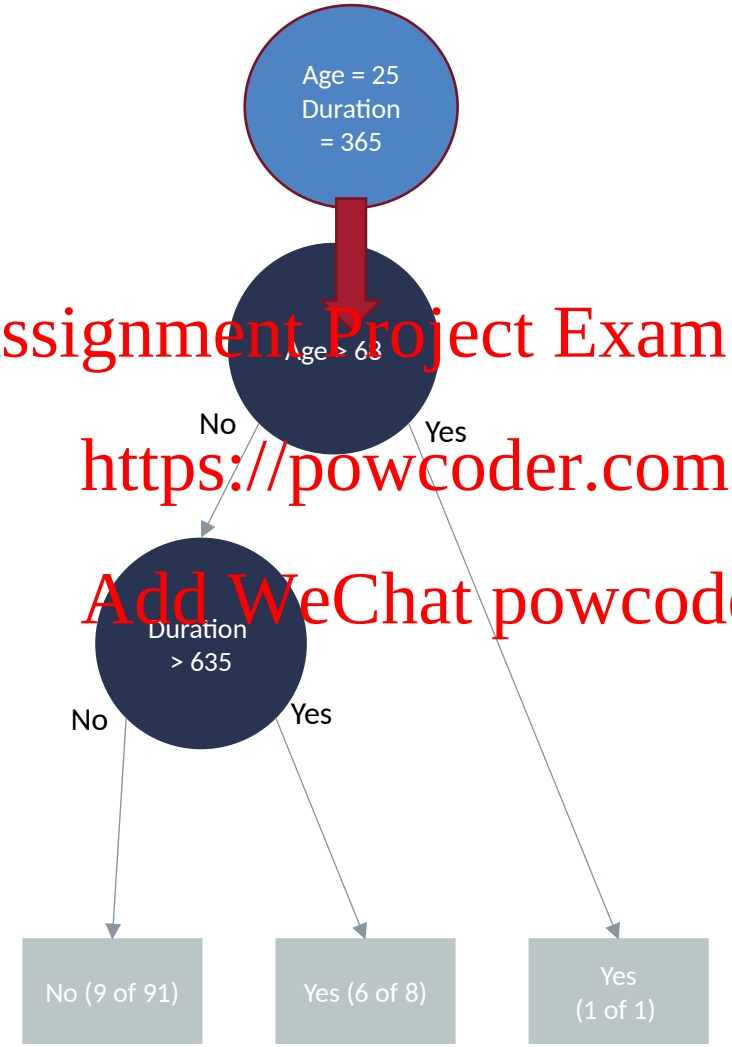
The duration is greater than 635 so the record lands at YES with a probability of 75% (6/8).



2 Rule Tree

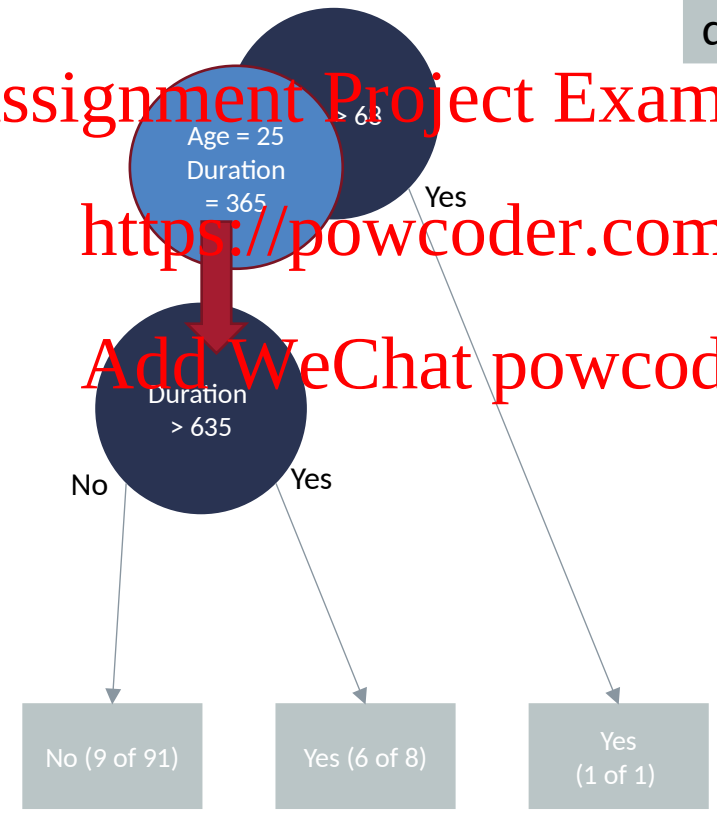


Let’s drop another record down our plinko tree

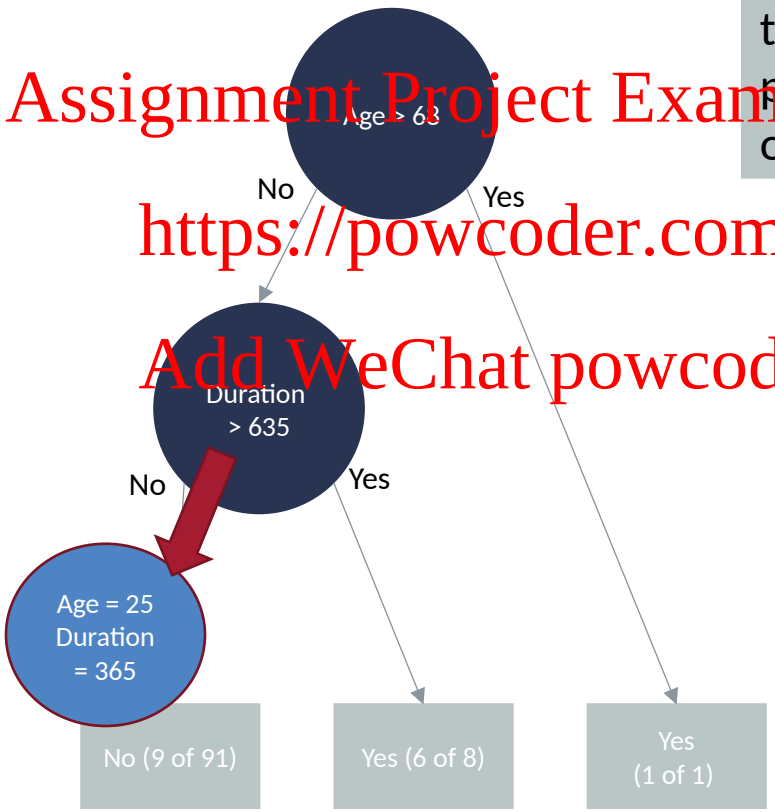


Let’s drop a new record down our plinko tree

The age was less than 68 so the record drops to the next decision point.

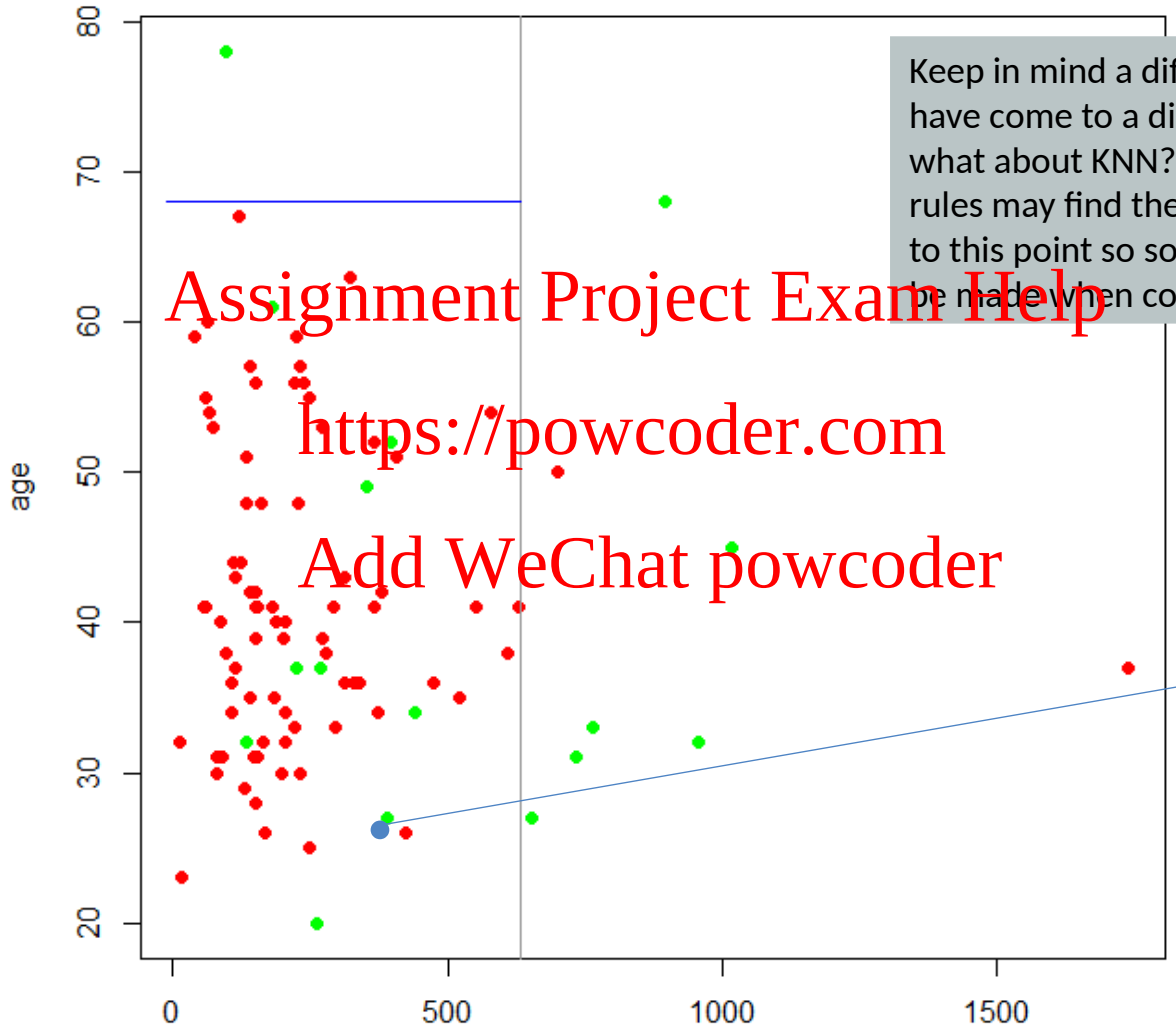


Let’s drop a new record down our plinko tree



Now the duration is less than 635 so the record drops to the terminal “NO” and has a probability of accepting the loan offer of ~10%

2 Rule Tree



Keep in mind a different algorithm may have come to a different conclusion… what about KNN? Also, adding more rules may find the local structure closer to this point so some judgment must still be made when constructing a tree.

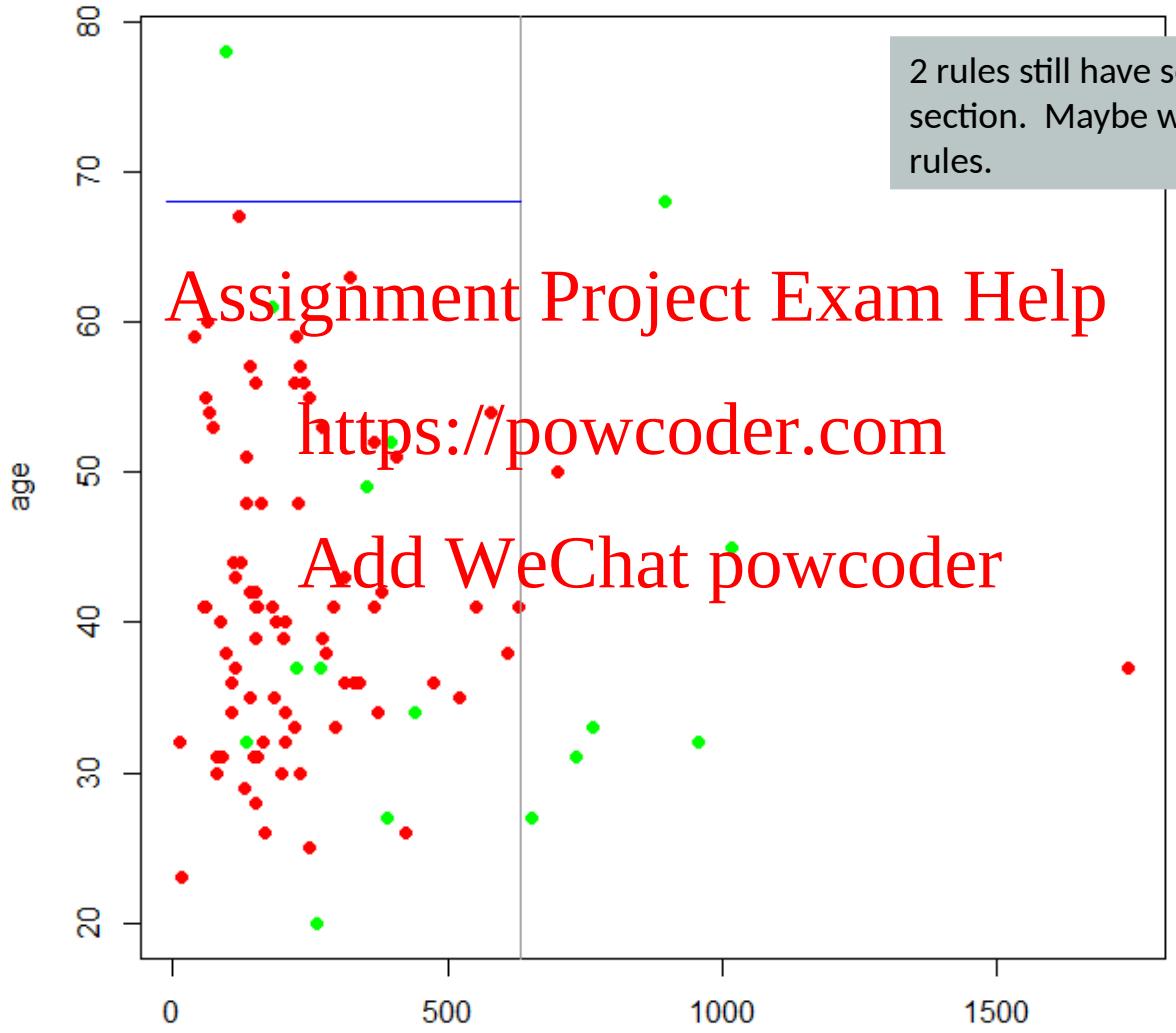
Age = 25
Duration
= 365

HARVARD UNIVERSITY

How a decision tree really splits data.

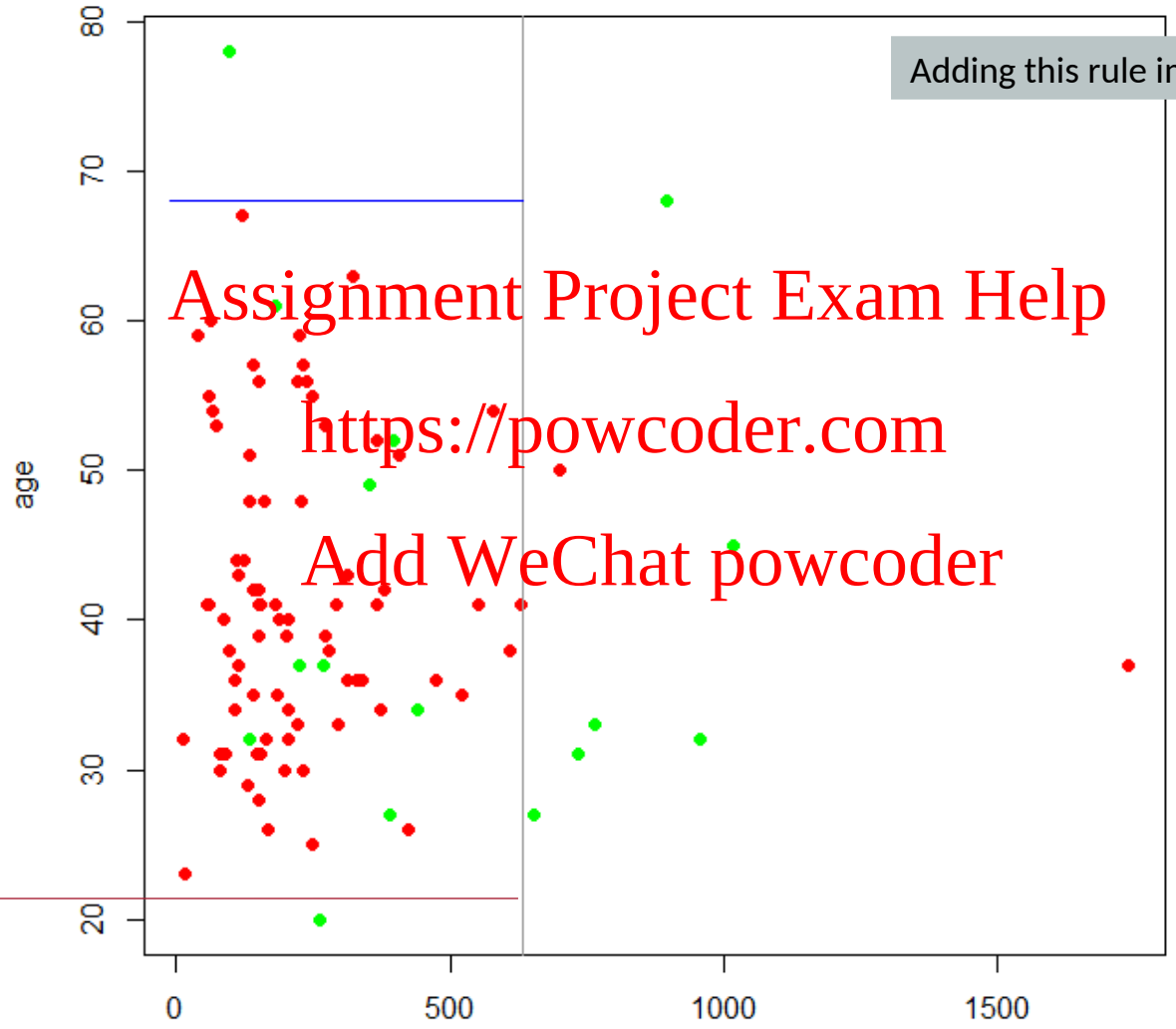
- Order records according to one variable, say duration
- Take a predictor value, say 600 (from the first record) and divide records into those with duration >= 600 and those < 600
- Measure resulting purity (homogeneity) of class in each resulting portion
- Try all other split values within the duration vector
- Repeat for other variable(s)
- Select the one variable & split that yields the most purity
- Since it splits at various values within a single vector, there is no need to standardize (center, scale, normalize).
- Complexity of the tree has to do with the number of layers allowed & the size, how many records, within each terminal node

An example of overfitting…

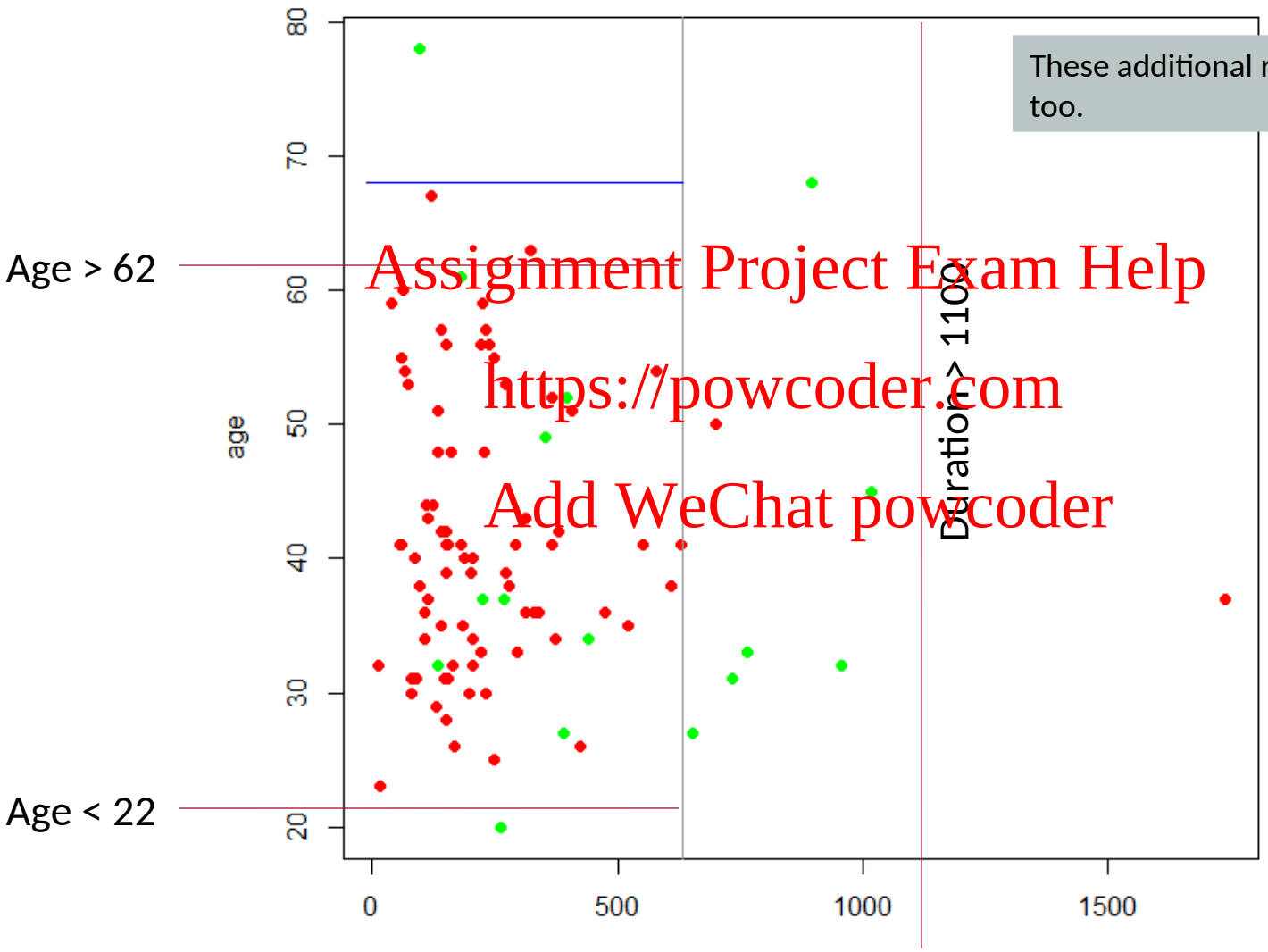


2 rules still have some impurity in each section. Maybe we should keep adding rules.

An example of overfitting…

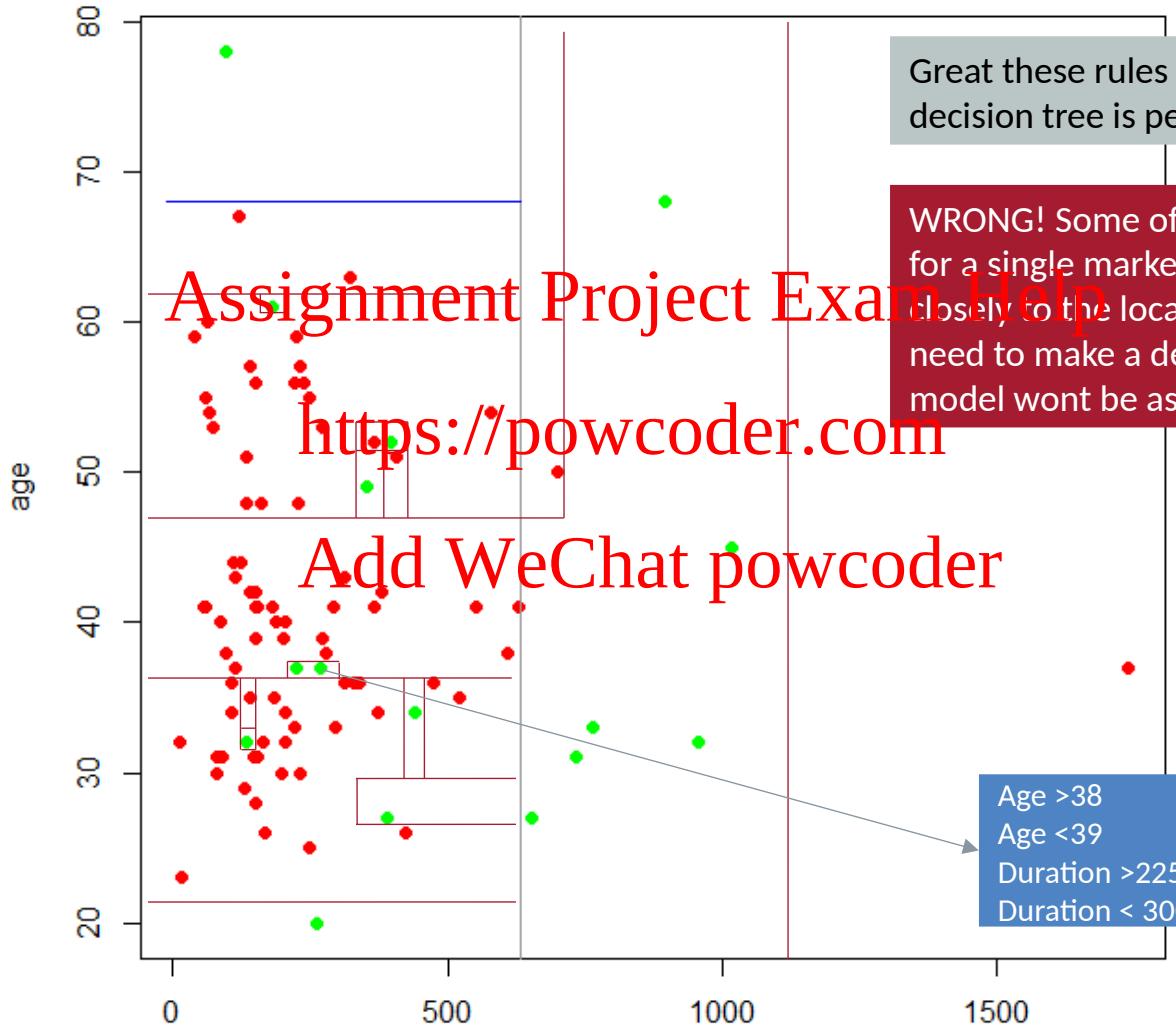


An example of overfitting…



These additional rules look like they help too.

An example of overfitting…



Great these rules are 100% pure! Our decision tree is perfect.

WRONG! Some of the rules were made for a single marketing offer. We fit to closely to the local structure so when we need to make a decision in the future this model wont be as accurate!

Age >38
Age <39
Duration >225
Duration < 300

Perfect node, but this is a rather specific customer case.

HARVARD UNIVERSITY

cp- complexity parameter

The cp value measures the size of the tree compared to its ability to separate the data. The tree will grow until the next split doesn't reduce the cp value…meaning that split added more complexity than is gained from the purity of the node.

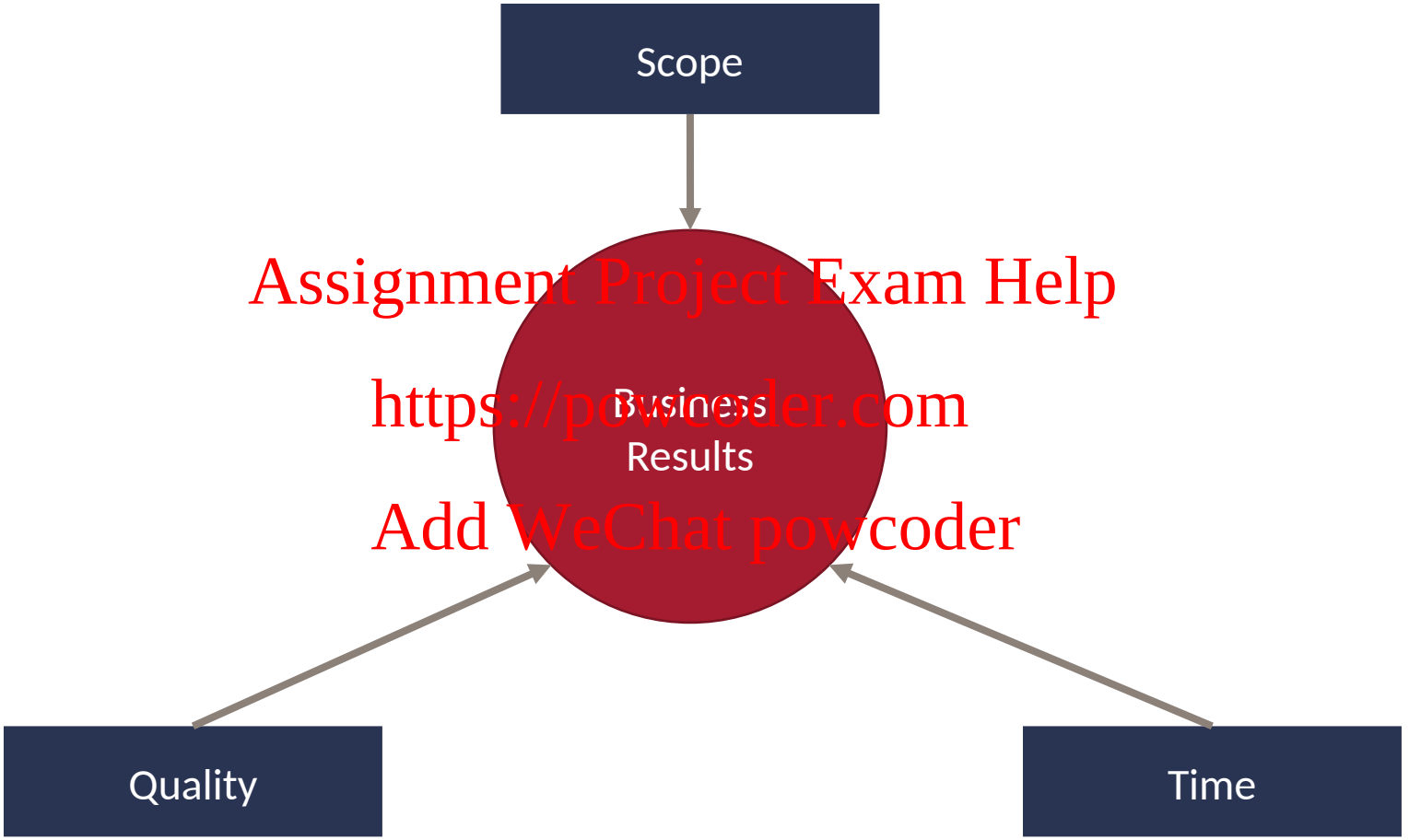
Setting cp to a negative amount ensures that the tree will be fully grown.

Open 1_Bank Loans Decision Tree.R

Agenda

Start	End	Item
		Decision Trees - explanation
		Decision Tree Example
		A to Z Decision Trees scripting example
		Break
		Random Forests

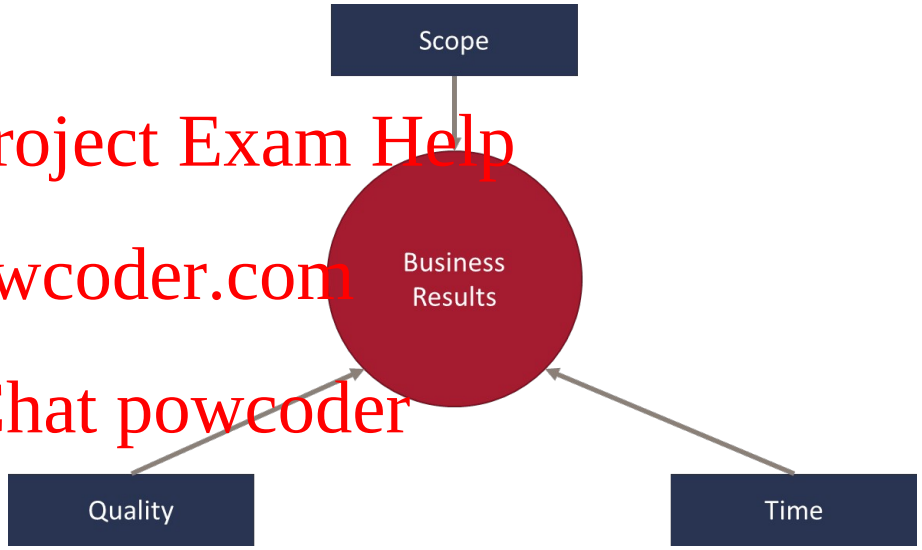
Framework for Business Results



Building a Bridge Example

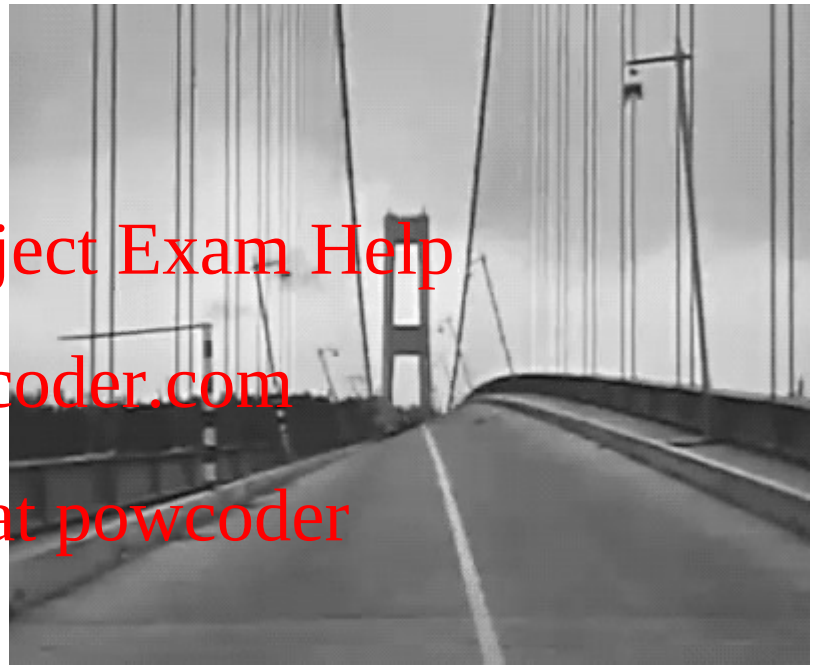
Scope

- From Point A to B
- Pavement
- Signs
- Traffic Divider
- Lane Markings
- Lighting
- Sidewalk
- Fencing
- Nets to catch suicides
- Flood Resistant
- Toll Booth
- Train underneath
- …



Building a Bridge Example

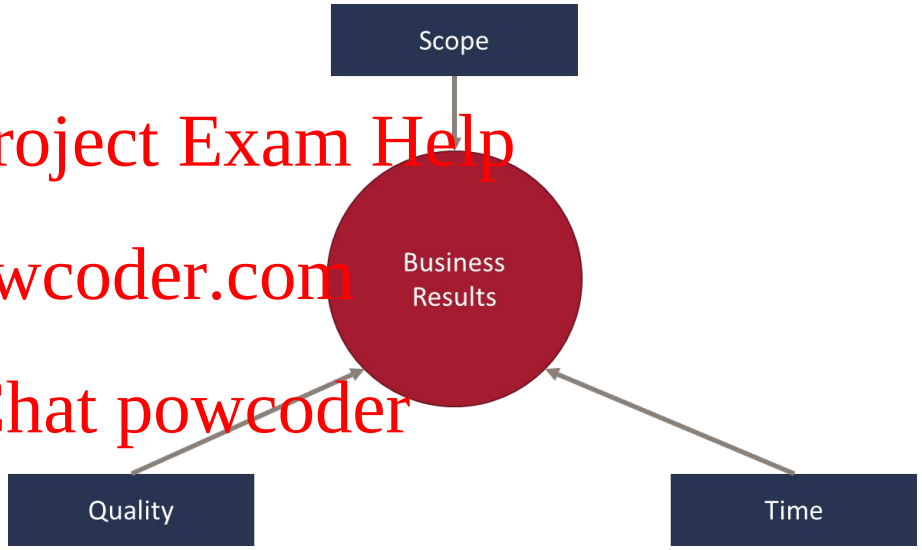
Quality
• Safe for cars, pedestrians • Stands up to inclement weather • Earth quake proof • Guaranteed for X years



Building a Bridge Example

Time

- Build it in 1 yr
- 2yrs
- 4 yrs – Golden Gate Bridge
- ??



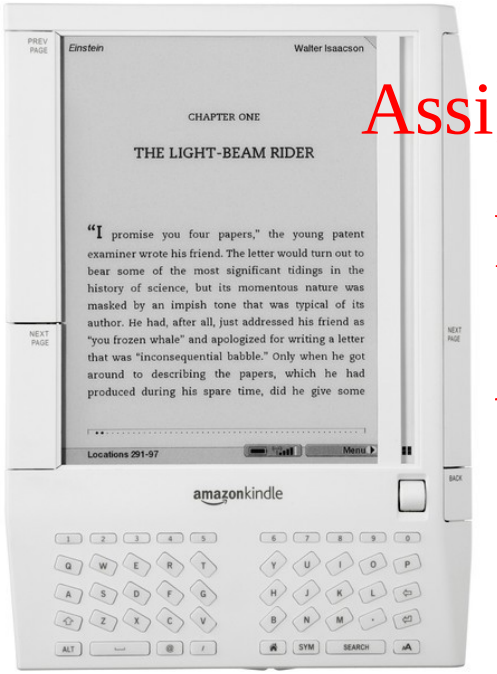
Business Projects get 2 of 3…choose wisely.

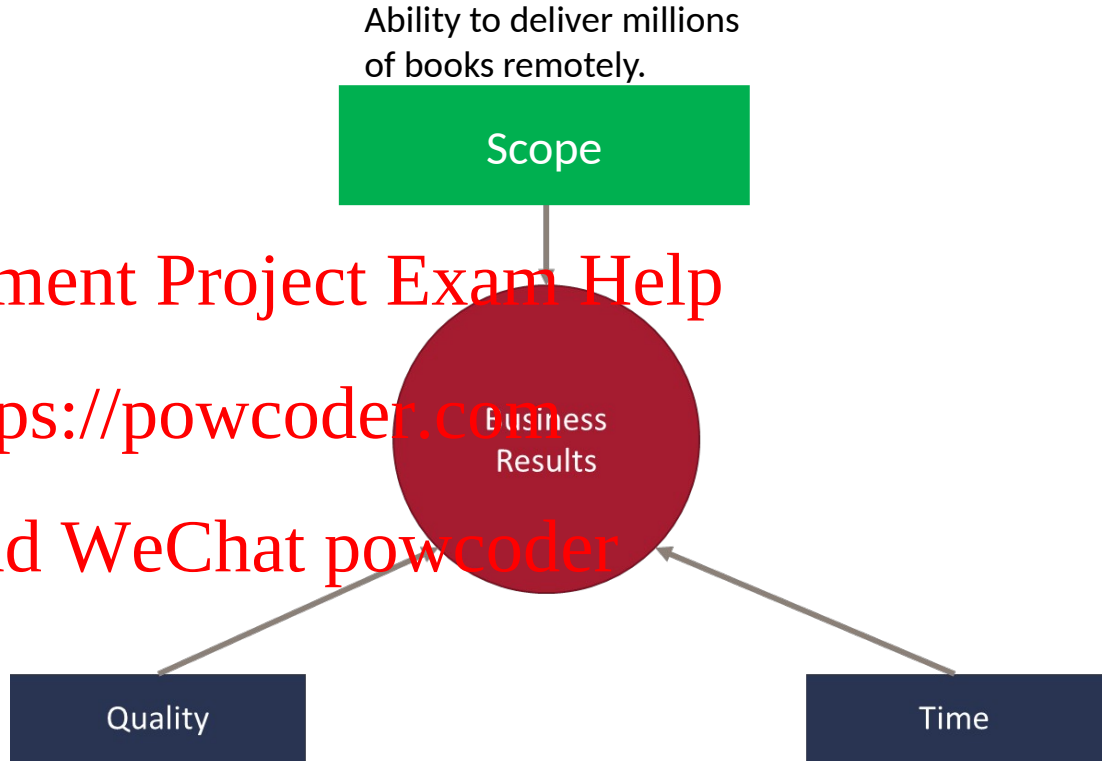
Quality

Scope

Time

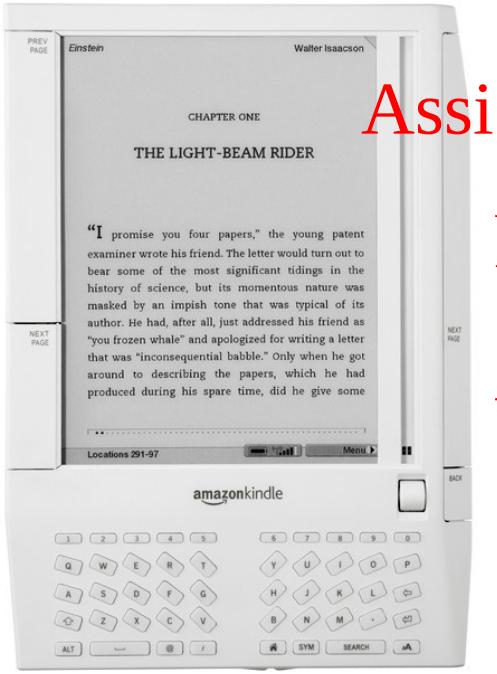
1st Generation Kindle

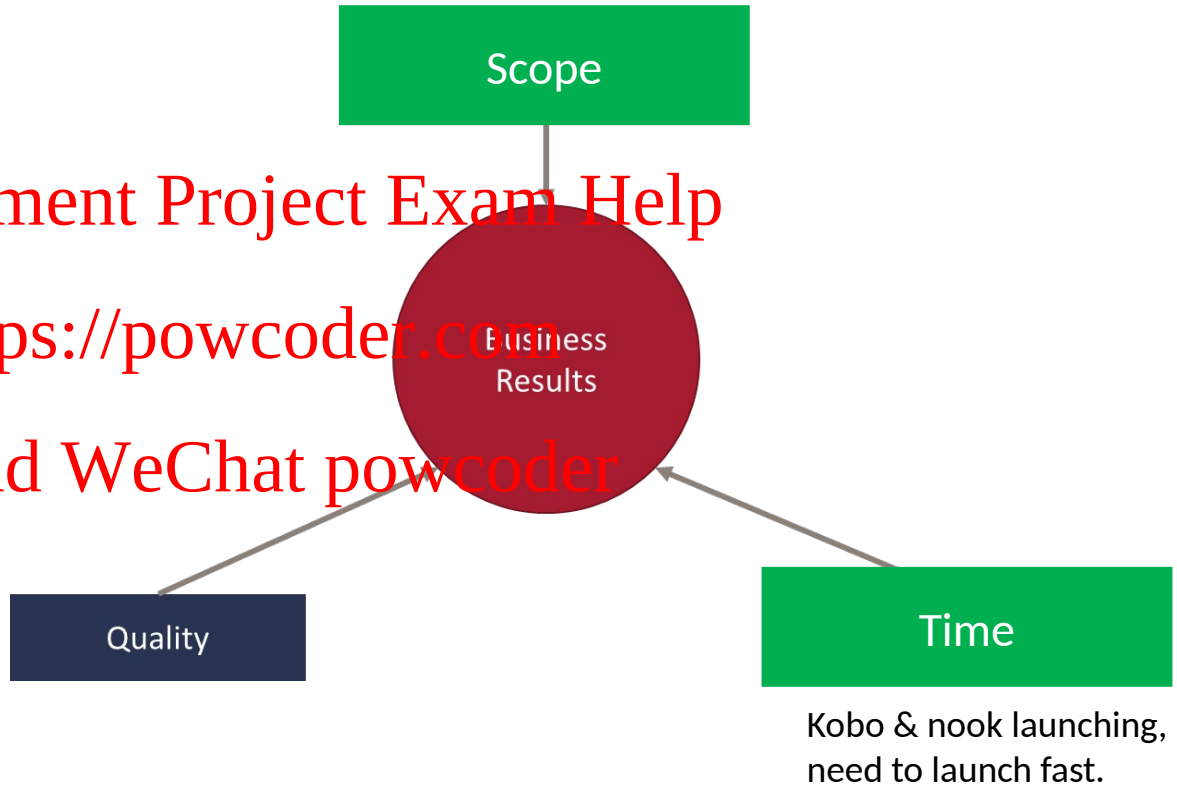




How good is an e-reader without the ability to deliver books remotely?

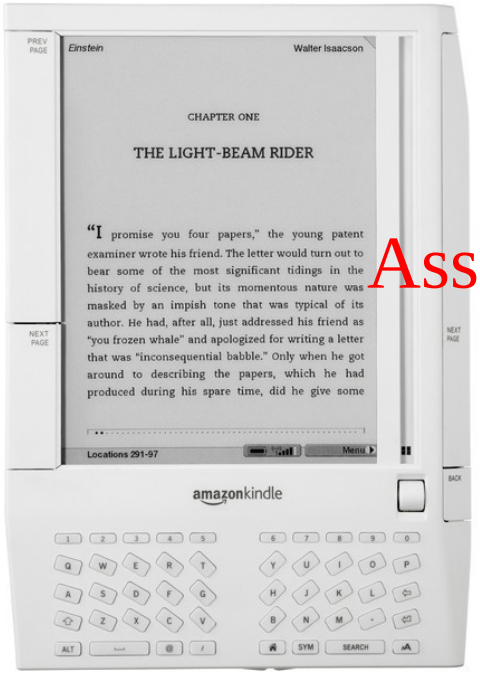
1st Generation Kindle

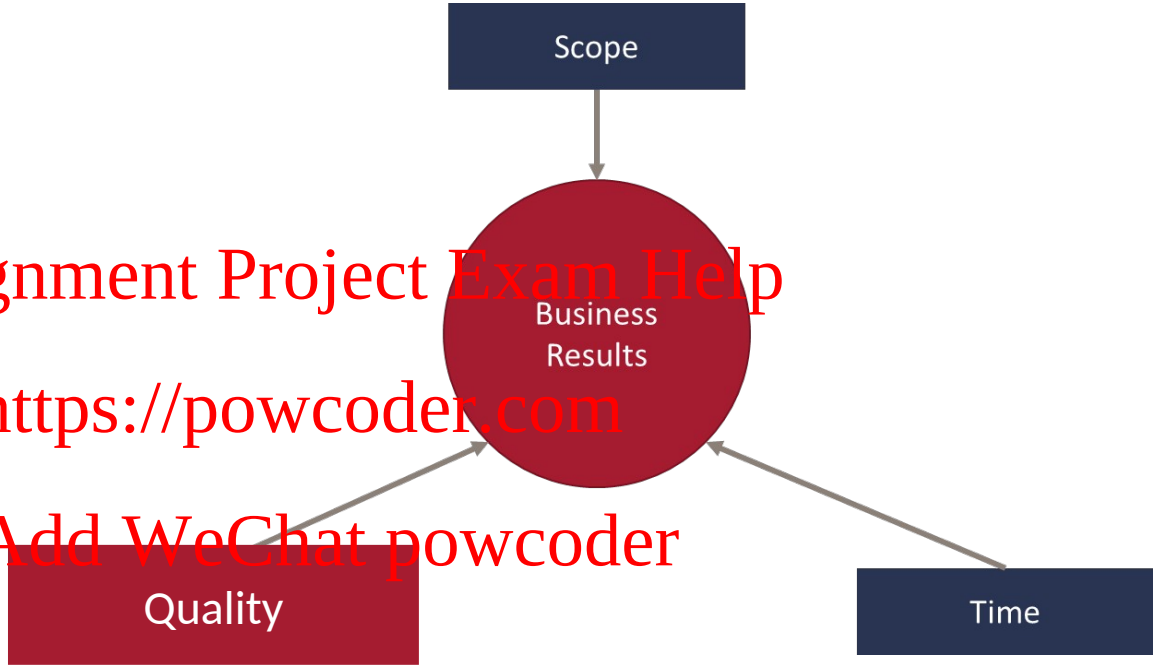




As a new market, Amazon wanted first mover advantage.

1st Generation Kindle





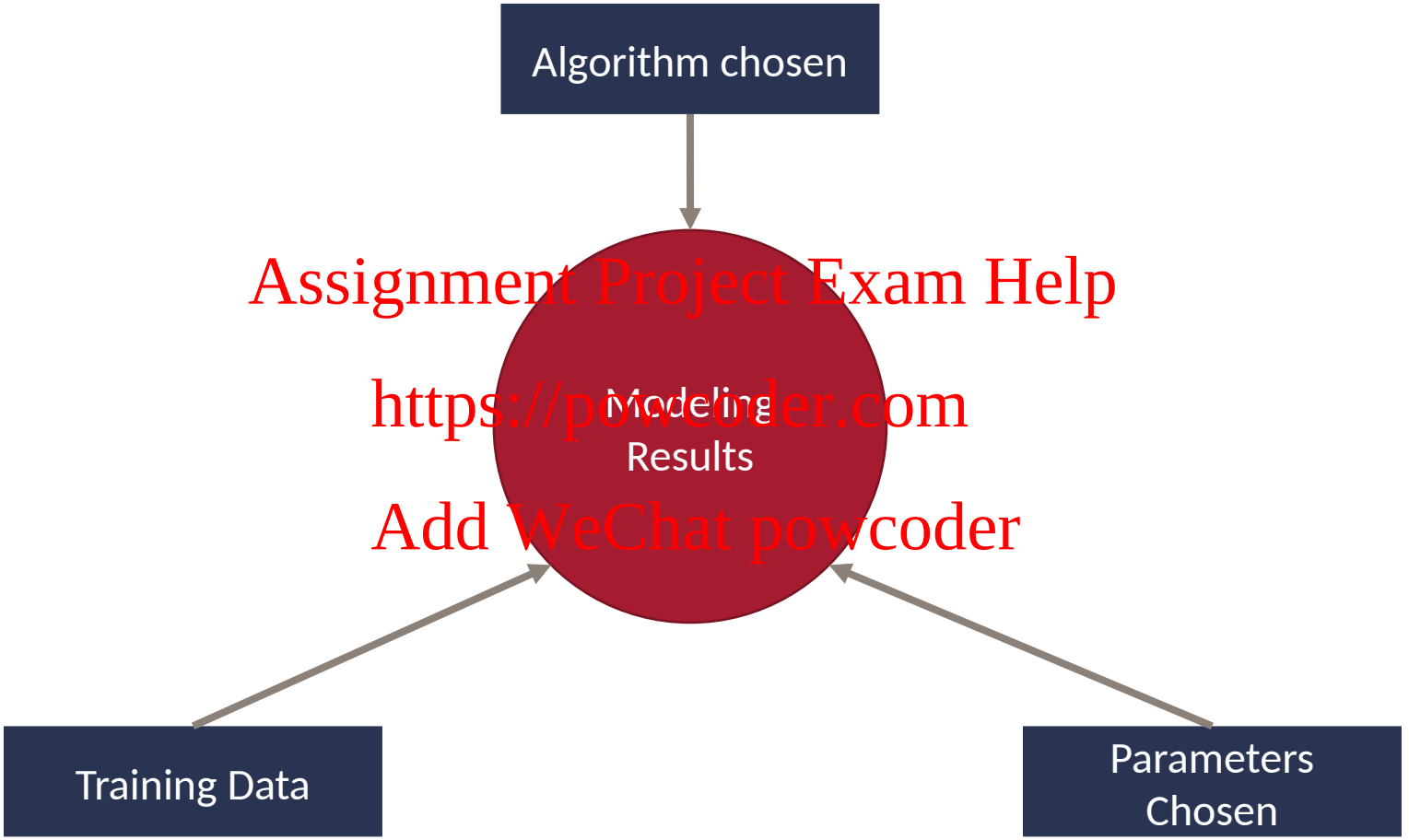
Scroll wheel & keyboard user interface breaks easily

Do current e-readers have a scroll wheel or keyboard?

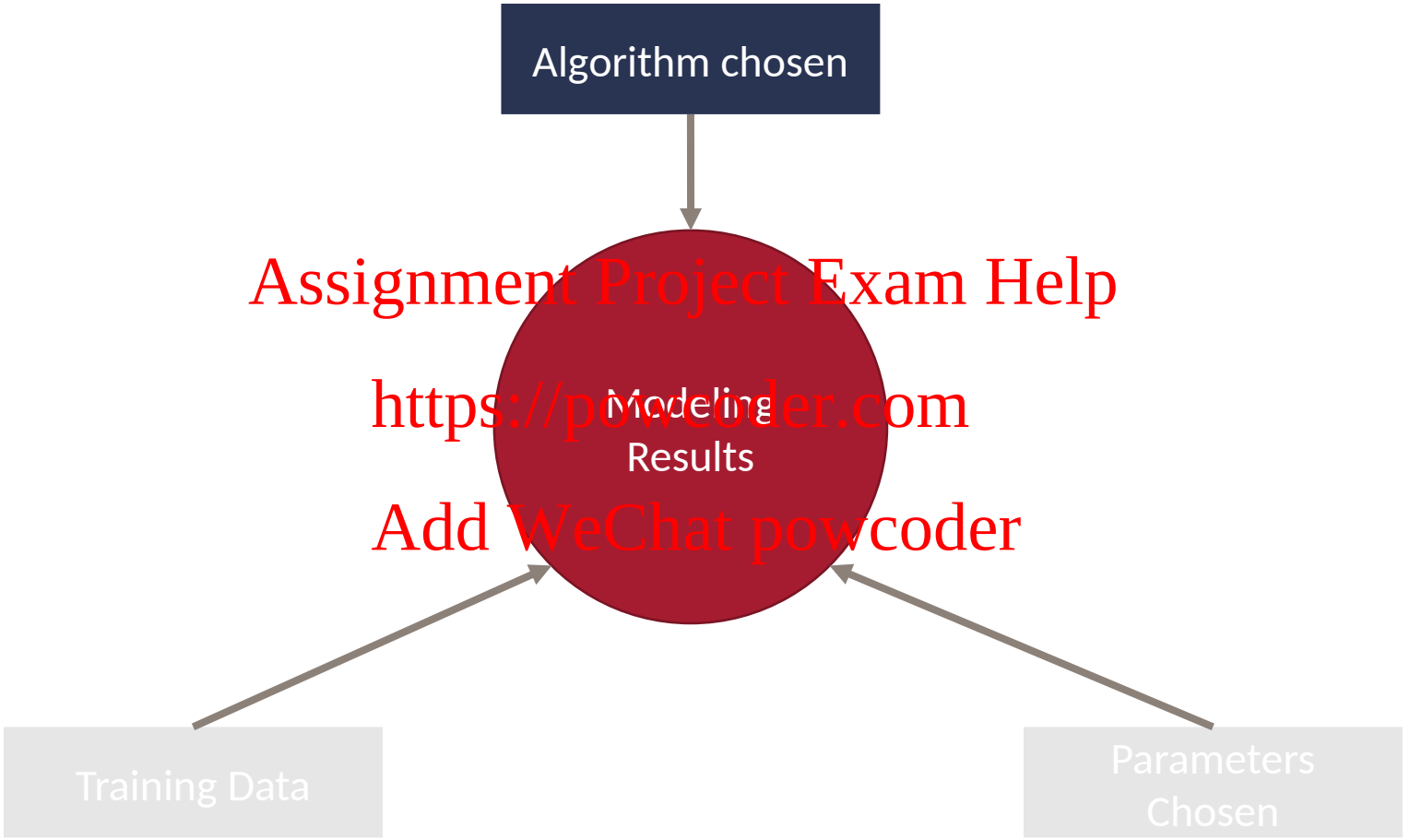
13yrs & 21 model changes…

https://techcrunch.com/2017/11/20/how-the-kindle-was-designed-through-10-years-and-15-generations/
https://en.wikipedia.org/wiki/Amazon_Kindle

In Data Science Modeling Results are similar



Modeling Results



Champion Vs Challenger!

Open
2_Bank Loans Algo Comparison.R

Agenda

Start	End	Item
		Decision Trees - explanation
		Decision Tree Example
		A to Z Decision Trees scripting example
		Break
		Random Forests

Random Forests…the Wisdom of the Crowd

1907 Vox Populi by Sir Francis Galton created the notion of "wisdom of crowds" as a phenomenon. It is the basis of modern search engines and crowdsourcing.

- Prevailing wisdom at the time was that crowds acted irrationally
  - Riots, mobs, cults
  - Galton wanted to explore trustworthiness and peculiarities of popular judgment
- Galton attended an Ox weight guessing competition at a cattle exhibition
  - Participants individually guessed a very wide weight range

Degrees of the length of Array 0°—100°	Estimates in lbs.	Centiles: Observed deviates from 1207 lbs.	Centiles: Normal p.e. = 37	Excess of Observed over Normal
5	1074	−133	−90	+43
10	1109	− 98	−70	+28
15	1126	− 81	−57	+24
20	1148	− 59	−46	+13
25	1162	− 45	−37	+ 8
30	1174	− 33	−29	+ 4
35	1181	− 26	−21	+ 5
40	1188	− 19	−14	+ 5
45	1197	− 10	− 7	+ 3
50	1207	0	0	0
55	1214	+ 7	+ 7	0
60	1219	+ 12	+14	− 2
65	1225	+ 18	+21	− 3
70	1230	+ 23	+29	− 6
75	1236	+ 29	+37	− 8
80	1243	+ 36	+46	−10
85	1254	+ 47	+57	−10
90	1267	+ 52	+70	−18
95	1293	+ 86	+90	− 4

q_1, q_3, the first and third quartiles, stand at 25° and 75° respectively.
m, the median or middlemost value, stands at 50°.
The dressed weight proved to be 1198 lbs.

The median weight was 9lbs (0.8%) off and was better than individual cattle experts.

Conditions for Wisdom of Crowds

Some machine learning methods behave like the cattle weight guessing participants.

For people:

- Each individual member or voter must have an independent source of information (examine the cattle for themselves)
- Make an individual guess not swayed by others (avoid group think or be blind to others' guesses)
- Mechanism must be in place to collate and organize the diverse votes.

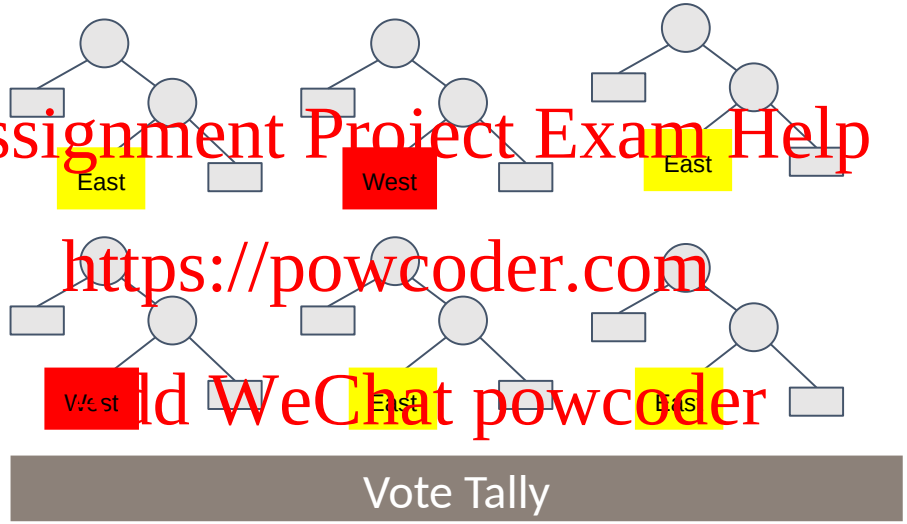
For algorithms:

- Each individual method needs to be blind to the others in construction
  - Each individual classification or vote is therefore independent
  - Aggregation/tabulation is easy in an computerized environment.

Many of today's best algorithms mimic a weak learner approach.

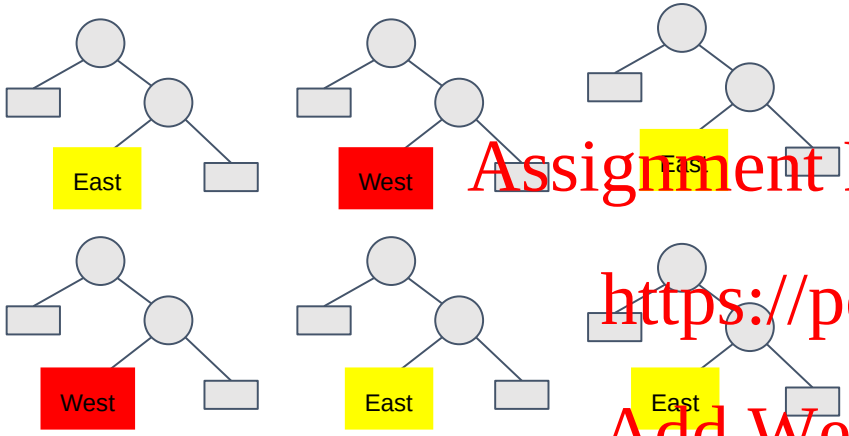
Random Forests

Each mini tree acts as a voter from the mob. Each voter is allowed to look at a random group of variables e.g. a mini data set to make its splits.



East	West
4	2

Random Forests

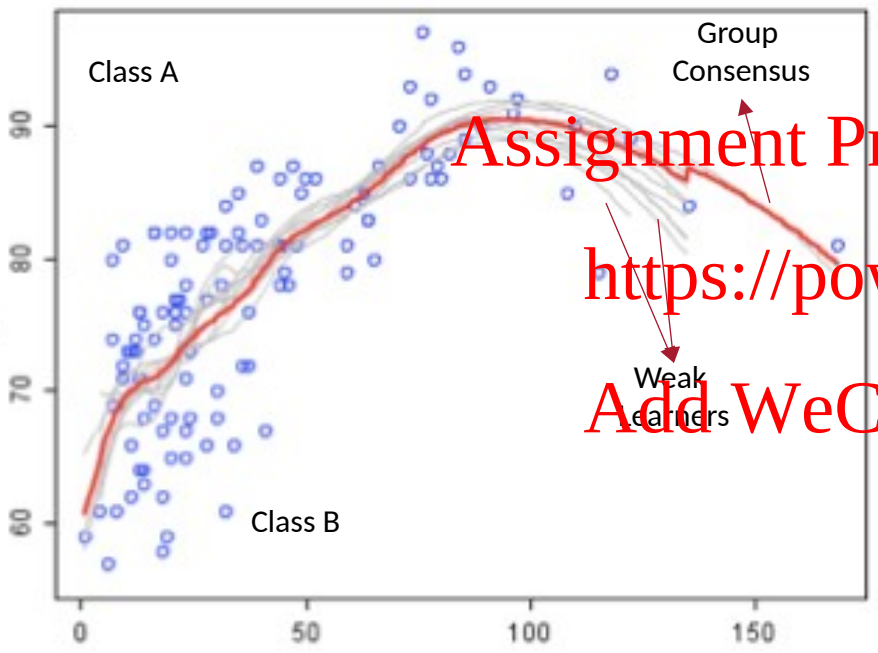


- This toy example shows 2 variables were selected for each voter from among beds, baths, sqft and price. An individual tree may choose beds/baths while another may get beds/sqft and still another gets sqft/price.
- A random forest grows many many classification trees
  - **Variable combinations are selected randomly** to create the forests but the number of variables or attributes needs to be specified
  - Each unknown value is then put through each tree to provide a final classification
  - Each tree then casts a vote for the unknown value.
- Each "dumb" learner gets a vote that is tabulated so the wisdom of the mob can be identified.

Since each tree focuses on a random combination of variables and random rows (bootstrap), some trees will be more accurate, others less so. In total the splits will identify most informative variables, the ones that are usually split upon and with enough trees the group wisdom is found.

Random Forests

Fictitious Visual



Data Science Considerations

- How many trees/voters?
- How many decision points (nodes)?
- How many features are allowed to be used in each tree build?
- How to account for missing values?
  - Imputation fills in but a tree can split on records that are missing. Thus vtreat's imputation with missing indicator as an appended variable may be a good alternative.

In this example, each decision tree tries to separate the data and by taking the most numerous outcome (tally) the best outcome is often found. Since they are weak learners (not a lot of depth and with fewer variables) among hundreds, overfitting is usually less of a concern.

How a random forest is really grown.

- Draw many re-samples of cases from the data (bootstrap)
- For each re-sampled set use a random subset of predictor variables to produce a tree. Some parameters include
  - Number of trees to grow – dictates the number of re-sampled sets
  - Number of predictor variables for each tree
- Combine the predictions/classifications from all the trees (the entire forest)
  - Votes are tallied for classification problems
  - Predictions are averaged for continuous problems
- Since it has many weak learners, overfitting is less of a concern. Some trees will get the unlikely outlier value while most won't.
- Since its fitting many hundreds of trees, it takes some time to train a model.

Open Bank Loans RandomForest REVISED.R

Summary

- Classification and Regression Trees are an easily understandable and transparent method for predicting or classifying new records
- A single tree is a graphical representation of a set of rules
- Tree growth must be stopped to avoid overfitting of the training data – cross-validation (CV) helps you pick the right level to stop tree growth – *will cover CV later in the course*
- Ensembles (random forests, boosting) improve predictive performance, but you lose interpretability and the rules embodied in a single tree

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)
- Linear Regression (continuous)
- Logistic Regression (Binary Classification)
- KNN (continuous & classification – binary or multi)

Today

- Decision Tree (classification, continuous)
- RandomForest (classification, continuous)

DT & RF are excellent modeling tools. Not bad for 6 classes!