

Introduction to Data Science

Lecture 05; April 27th, 2015

Ernst Henle

ErnstHe@UW.edu

Skype: ernst.predixion

Agenda



- Social Interactions
 - LinkedIn
 - Encourage Group Homework
 - Get help early
- Announcements of Guest Lectures
- Review Homework
- In-Class Exercise and Quiz 05 (Elbow.R)
- Data and Models in Supervised Learning (continued from last week)
- Break
- Schema for Classification (Next week's Quiz 06a)
- Break
- Partition Modeling Data (Homework Assignment)
- “Real World” Classification (Homework Assignment)

Announcements

- Guest Lecture: May 4th 1-hour by Marius Marcu “Business Aspects of Data Science”
- Guest Lecture: May 11th 1-hour by Ben Olsen on “Design Concepts for Visualization”

Review: K-Means in R

- KMeans.R
- KMeansHelper.R
- TestKMeans.R



**99 little bugs in the code.
99 little bugs in the code.
Take one down, patch it around.**

127 little bugs in the code...

Quiz 05a

- Link: <https://catalyst.uw.edu/webq/survey/ernsthe/269279>
- Start R. Use Elbow.R and CollegePlans.csv

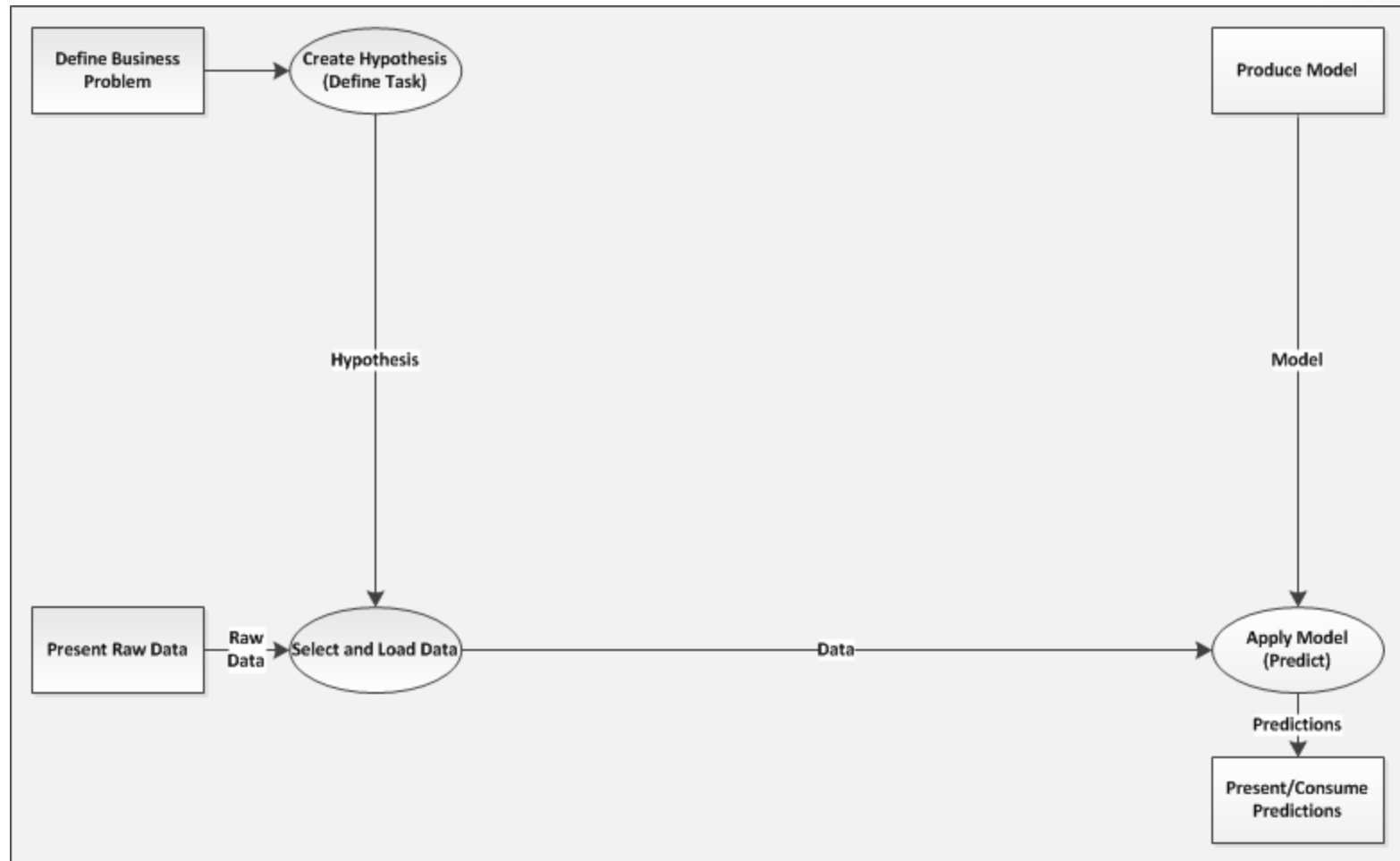
Data and Models in Supervised Learning

(1) Model Acts on Data



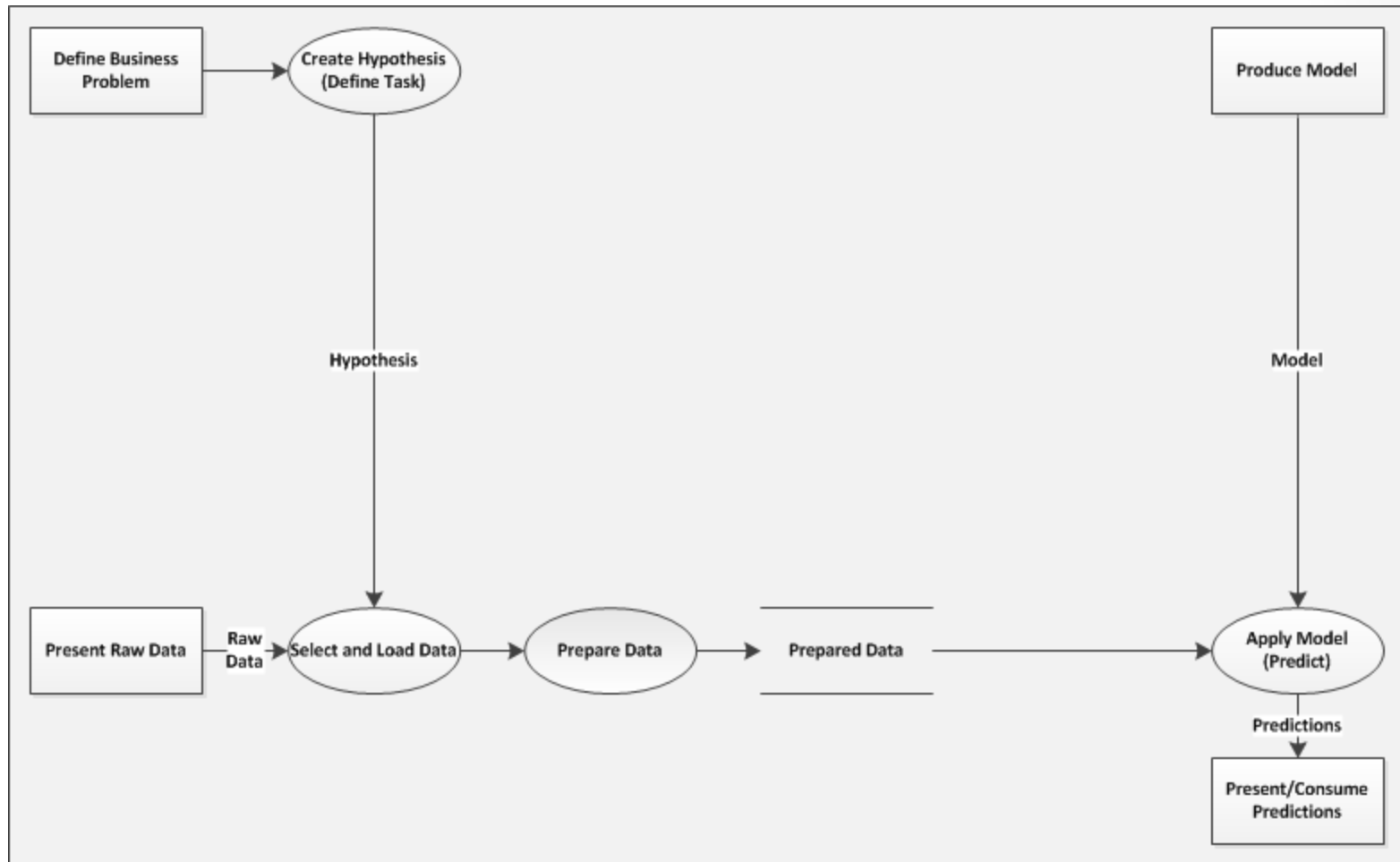
Model + Data → Prediction

(2) Data Selection Reflects Hypothesis / Business Problem



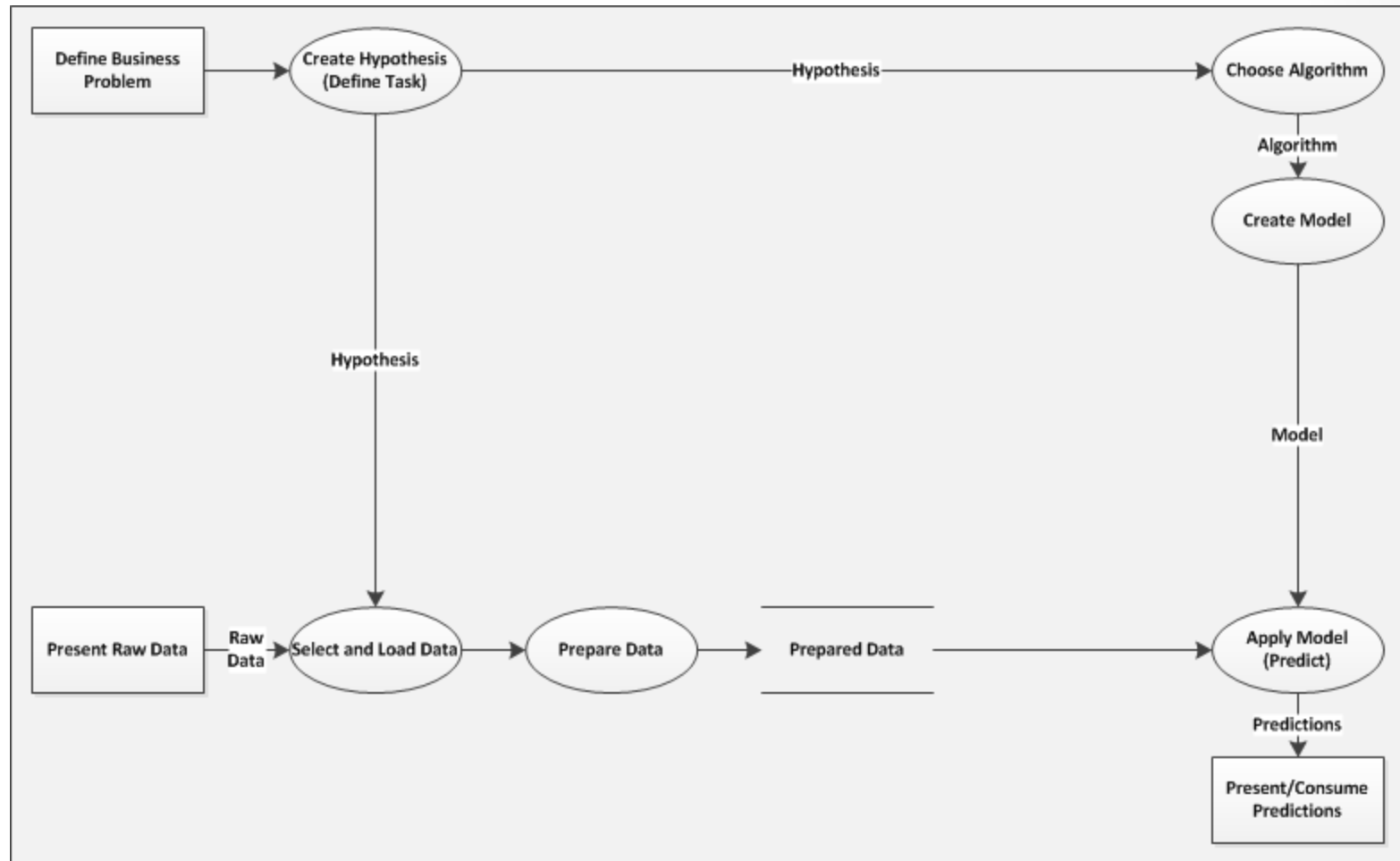
Hypothesis determines what data are loaded

(3) Data Needs Preparation



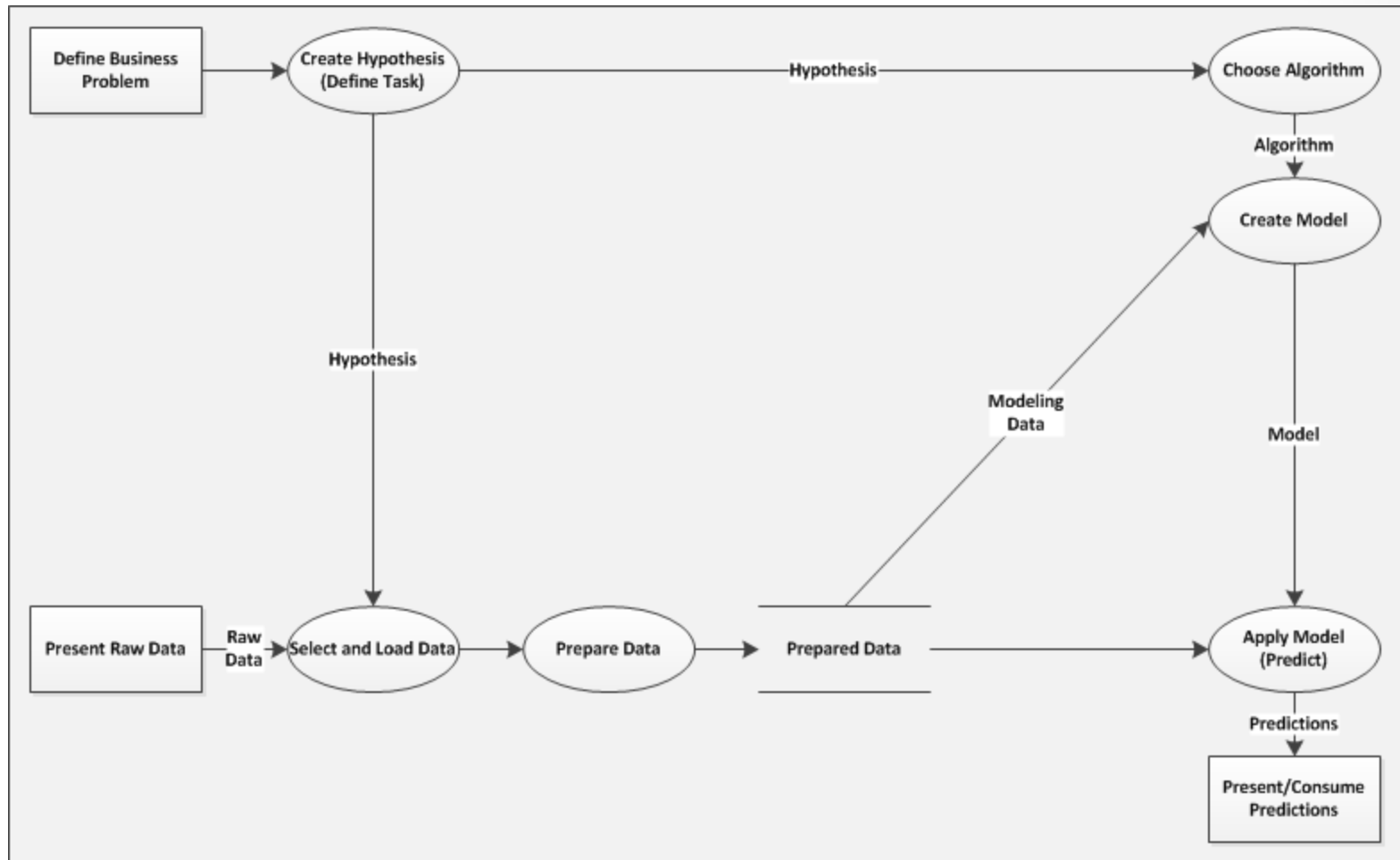
Data need to be prepared for use by a model.

(4) Model Creation Reflects Hypothesis / Business Problem



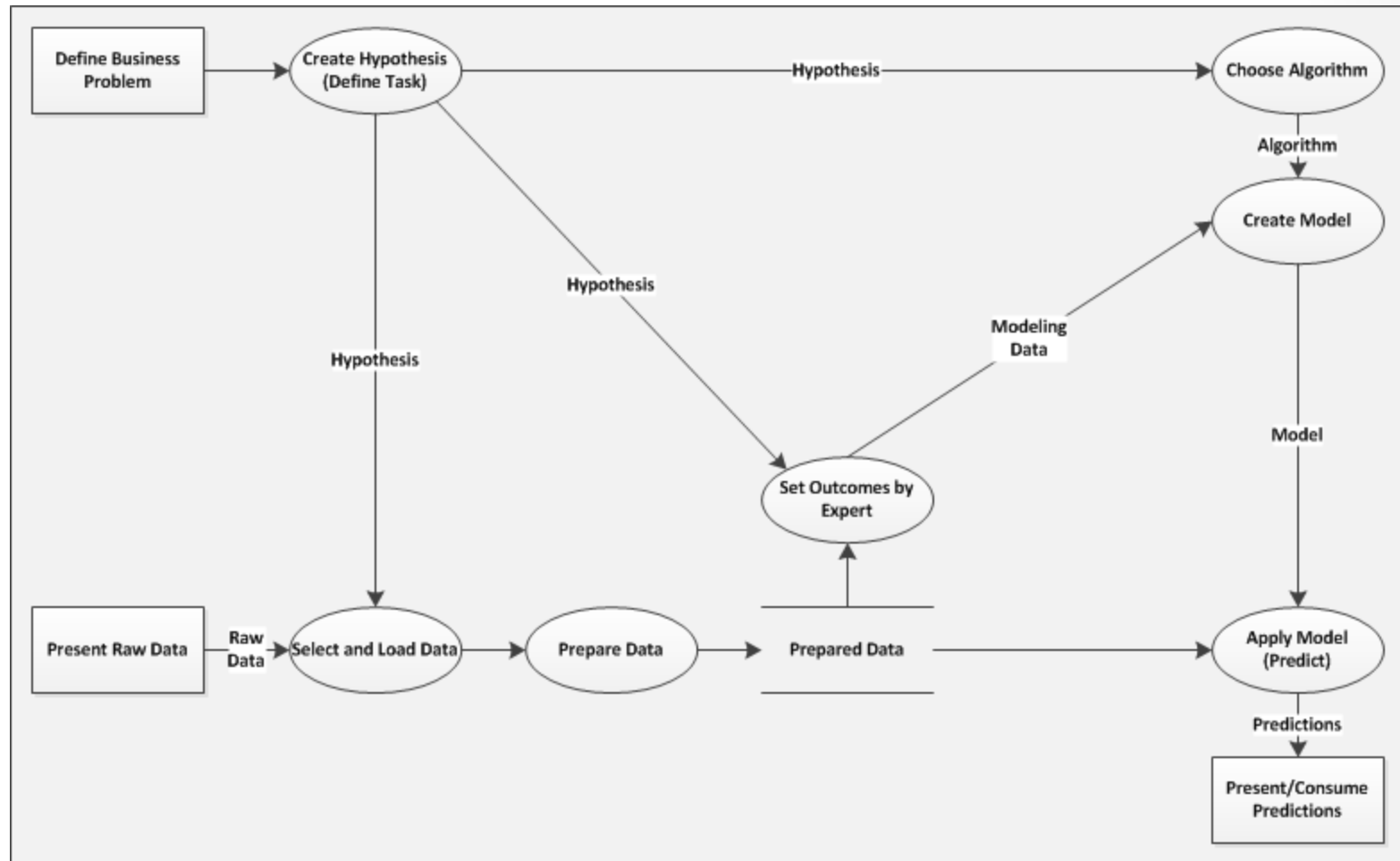
Hypothesis determines the choice of Algorithm.

(5) Model Creation needs Data



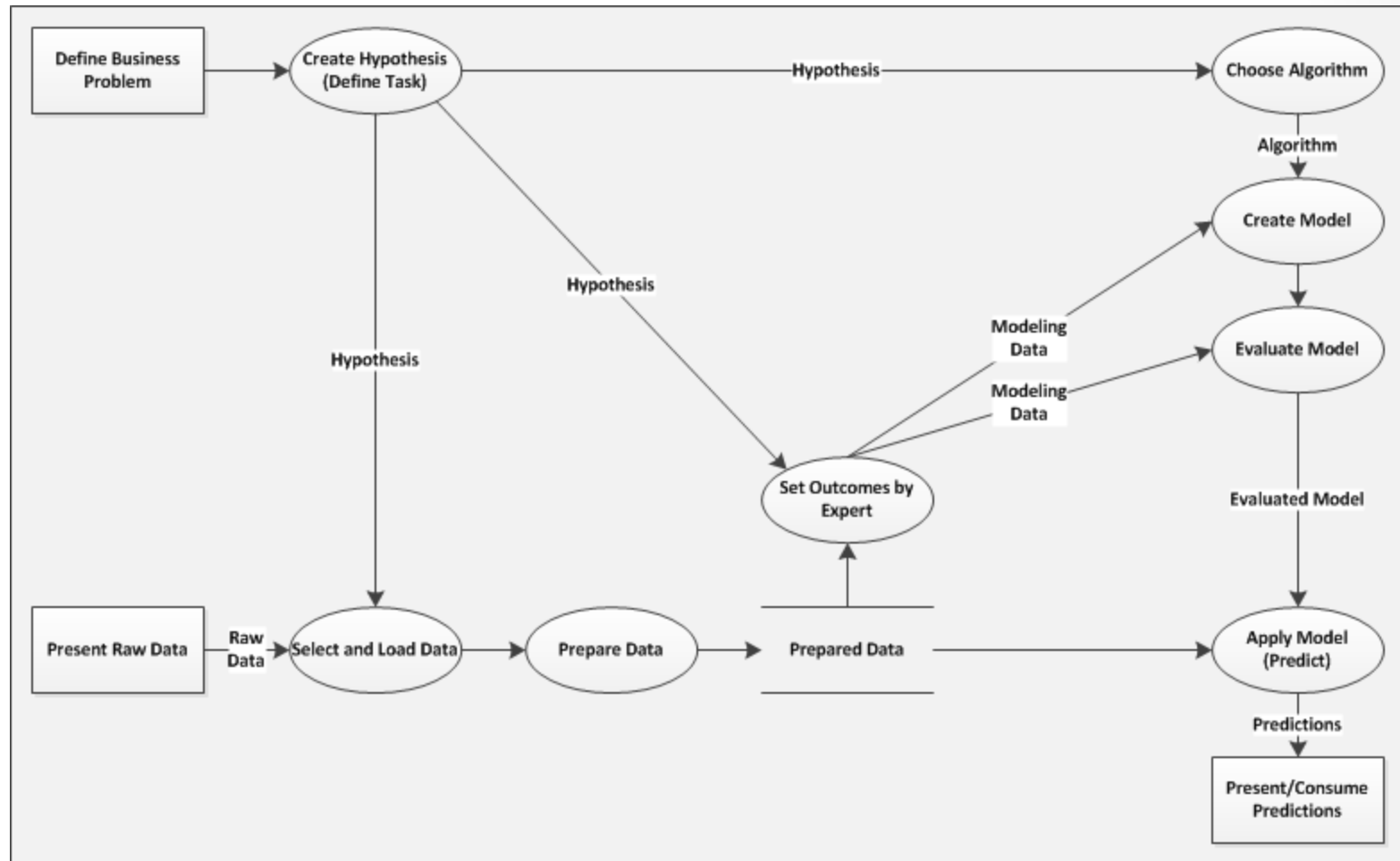
Data + Algorithm → Model

(6) Supervised Training needs Data Labeled with Outcomes



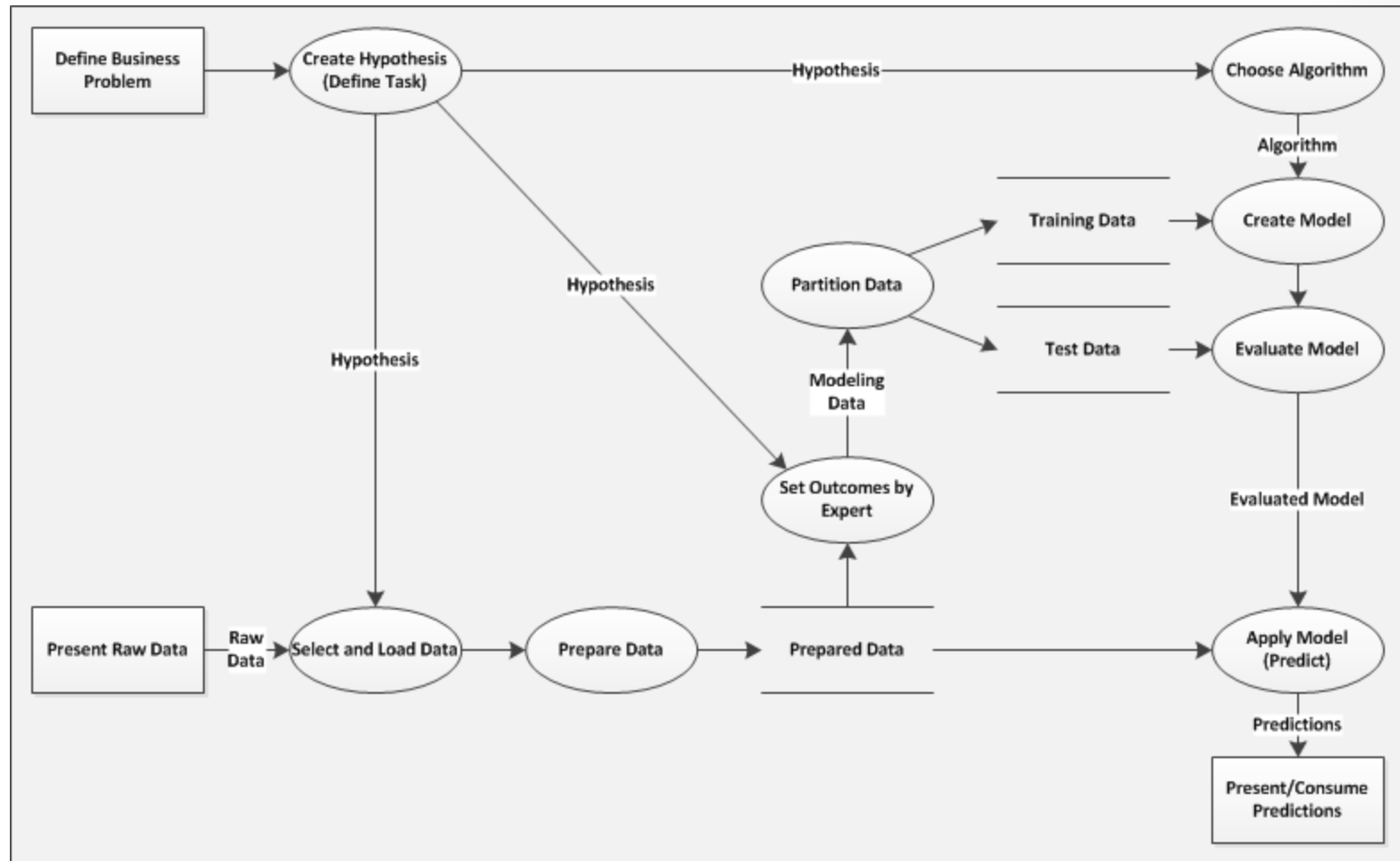
Supervised Learning requires expert labeling of data.

(7) Models need to be Evaluated



Do not trust predictions from an un-tested model!

(8) Creation & Evaluation of Model may not use same Data



Do not test a model using training data!

Data and Models in Supervised Learning

Break

Classification Schema

Classification Schema (0)

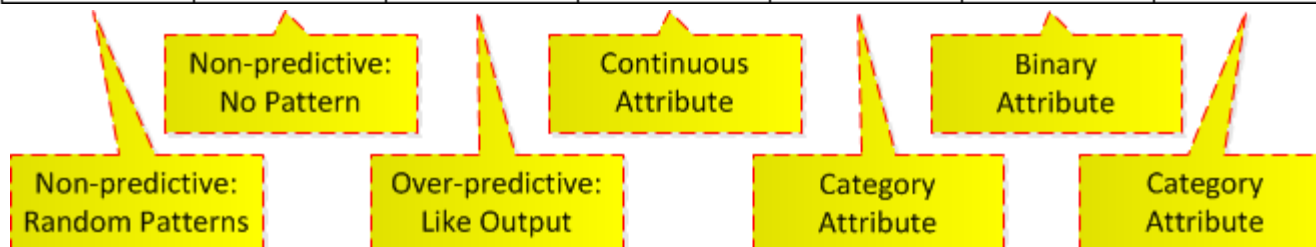
- Rectangular Modeling Dataset
 - Schema
 - Input columns
 - Output column (target column, outcome)
 - Classification: Category Column
 - Partition For Training and Test Data
 - Incremental Data
- Algorithm
 - Classification
 - Logistic Regression
 - Neural Network
 - Decision Tree
 - Naïve Bayes

Classification Schema (1)

Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5	Attribute 6	Attribute 7
330-272-449	Seaborg	Good	0.123	red	1	Yes
330-272-450	Seaborg	Bad	0.987	green	1	No
330-272-451	Seaborg	Yes	0.245	blue	0	Yes
720-273-500	Seaborg	Yes	0.254	blue	1	Yes
720-273-501	Seaborg	Bad	0.244	blue	0	No
720-273-502	Seaborg		0.415	green	0	Maybe
110-272-461	Seaborg	Yes	0.925	red	1	Yes
110-272-462	Seaborg	Yes	0.376	green	0	Yes
220-273-700	Seaborg	Bad	0.615	green	1	No
220-274-701	Seaborg		0.321	blue	0	Maybe
220-275-703	Seaborg	Bad	0.098	green	0	No
220-275-704	Seaborg	Bad	0.765	red	1	No

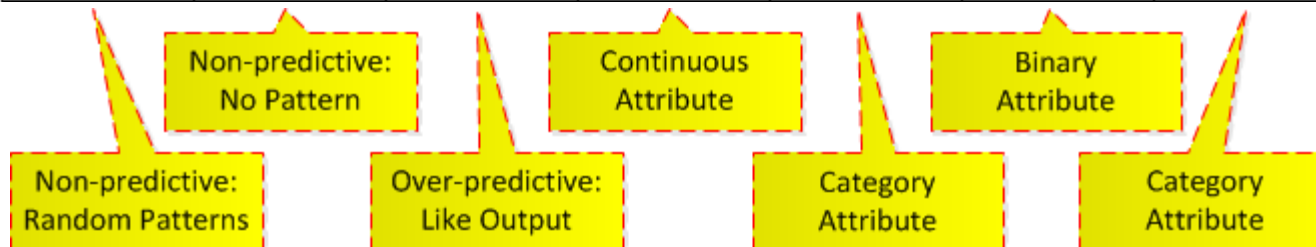
Classification Schema (2)

Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5	Attribute 6	Attribute 7
330-272-449	Seaborg	Good	0.123	red	1	Yes
330-272-450	Seaborg	Bad	0.987	green	1	No
330-272-451	Seaborg	Yes	0.245	blue	0	Yes
720-273-500	Seaborg	Yes	0.254	blue	1	Yes
720-273-501	Seaborg	Bad	0.244	blue	0	No
720-273-502	Seaborg		0.415	green	0	Maybe
110-272-461	Seaborg	Yes	0.925	red	1	Yes
110-272-462	Seaborg	Yes	0.376	green	0	Yes
220-273-700	Seaborg	Bad	0.615	green	1	No
220-274-701	Seaborg		0.321	blue	0	Maybe
220-275-703	Seaborg	Bad	0.098	green	0	No
220-275-704	Seaborg	Bad	0.765	red	1	No



Classification Schema (3)

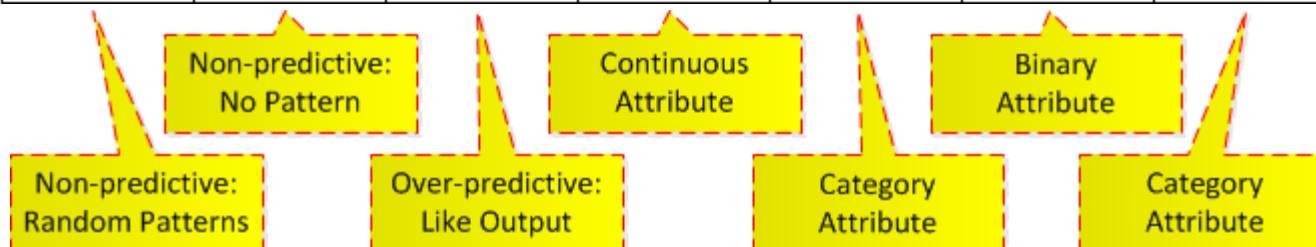
Key Column	Constant	Proxy Column	Input Column 1	Input Column 2	Input Column 3	Outcome Column
330-272-449	Seaborg	Good	0.123	red	1	Yes
330-272-450	Seaborg	Bad	0.987	green	1	No
330-272-451	Seaborg	Yes	0.245	blue	0	Yes
720-273-500	Seaborg	Yes	0.254	blue	1	Yes
720-273-501	Seaborg	Bad	0.244	blue	0	No
720-273-502	Seaborg		0.415	green	0	Maybe
110-272-461	Seaborg	Yes	0.925	red	1	Yes
110-272-462	Seaborg	Yes	0.376	green	0	Yes
220-273-700	Seaborg	Bad	0.615	green	1	No
220-274-701	Seaborg		0.321	blue	0	Maybe
220-275-703	Seaborg	Bad	0.098	green	0	No
220-275-704	Seaborg	Bad	0.765	red	1	No



Classification Schema (4)

Outcome ~ Column 1 + Column 2 + Column 3

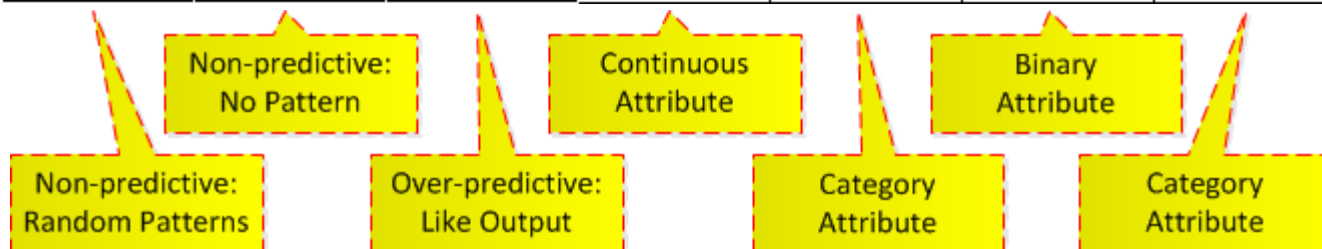
Key Column	Constant	Proxy Column	Input Column 1	Input Column 2	Input Column 3	Outcome Column
330-272-449	Seaborg	Good	0.123	red	1	Yes
330-272-450	Seaborg	Bad	0.987	green	1	No
330-272-451	Seaborg	Yes	0.245	blue	0	Yes
720-273-500	Seaborg	Yes	0.254	blue	1	Yes
720-273-501	Seaborg	Bad	0.244	blue	0	No
720-273-502	Seaborg		0.415	green	0	Maybe
110-272-461	Seaborg	Yes	0.925	red	1	Yes
110-272-462	Seaborg	Yes	0.376	green	0	Yes
220-273-700	Seaborg	Bad	0.615	green	1	No
220-274-701	Seaborg		0.321	blue	0	Maybe
220-275-703	Seaborg	Bad	0.098	green	0	No
220-275-704	Seaborg	Bad	0.765	red	1	No



Classification Schema (5)

Outcome ~ Column 1 + Column 2 + Column 3

			Input Column 1	Input Column 2	Input Column 3	Outcome Column
			0.123	red	1	Yes
			0.987	green	1	No
			0.245	blue	0	Yes
			0.254	blue	1	Yes
			0.244	blue	0	No
			0.415	green	0	Maybe
			0.925	red	1	Yes
			0.376	green	0	Yes
			0.615	green	1	No
			0.321	blue	0	Maybe
			0.098	green	0	No
			0.765	red	1	No



Classification Schema (6)

Outcome ~ Column 1 + Column 2 + Column 3

Input Column 1	Input Column 2	Input Column 3	Outcome Column
0.123	red	1	Yes
0.987	green	1	No
0.245	blue	0	Yes
0.254	blue	1	Yes
0.244	blue	0	No
0.415	green	0	Maybe
0.925	red	1	Yes
0.376	green	0	Yes
0.615	green	1	No
0.321	blue	0	Maybe
0.098	green	0	No
0.765	red	1	No

Continuous
Attribute

Binary
Attribute

Category
Attribute

Category
Attribute

Classification Schema (7)

Outcome ~ Column 1 + Column 2 + Column 3

Modeling Data (100-50000 rows)	Input Column 1	Input Column 2	Input Column 3	Outcome Column
	0.123	red	1	Yes
	0.987	green	1	No
	0.245	blue	0	Yes
	0.254	blue	1	Yes
	0.244	blue	0	No
	0.415	green	0	Maybe
	0.925	red	1	Yes
	0.376	green	0	Yes
	0.615	green	1	No
	0.321	blue	0	Maybe
	0.098	green	0	No
	0.765	red	1	No

Classification Schema (8)

Outcome ~ Column 1 + Column 2 + Column 3

		Input Column 1	Input Column 2	Input Column 3	Outcome Column
Training Data (50-50000 rows)	Modeling Data (100-50000 rows)	0.123	red	1	Yes
		0.987	green	1	No
		0.245	blue	0	Yes
		0.254	blue	1	Yes
		0.244	blue	0	No
		0.415	green	0	Maybe
		0.925	red	1	Yes
		0.376	green	0	Yes
		0.615	green	1	No
		0.321	blue	0	Maybe
		0.098	green	0	No
		0.765	red	1	No

Classification Schema (9)

Outcome ~ Column 1 + Column 2 + Column 3

		Input Column 1	Input Column 2	Input Column 3	Outcome Column
Test Data (50-5000 rows)	Modeling Data (100-50000 rows)	0.123	red	1	Yes
		0.987	green	1	No
		0.245	blue	0	Yes
		0.254	blue	1	Yes
		0.244	blue	0	No
		0.415	green	0	Maybe
		0.925	red	1	Yes
		0.376	green	0	Yes
		0.615	green	1	No
		0.321	blue	0	Maybe
		0.098	green	0	No
		0.765	red	1	No

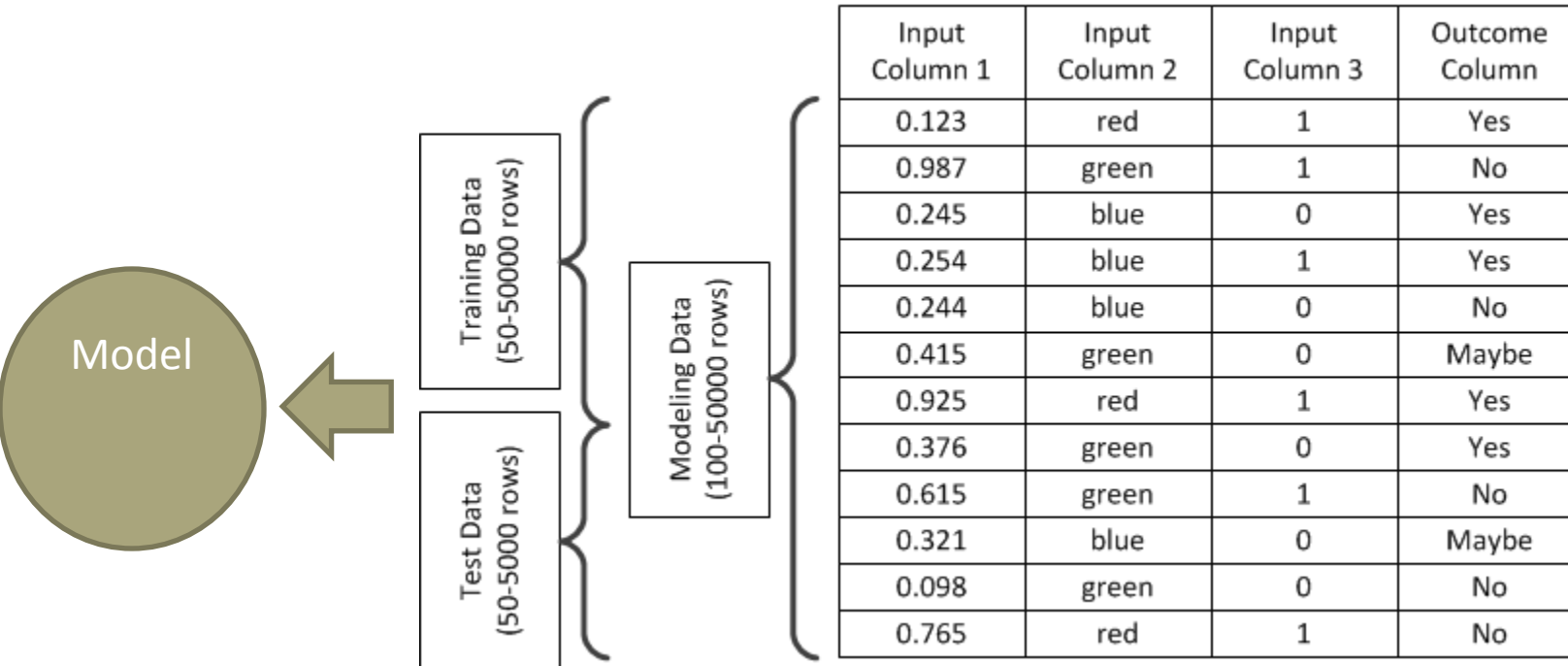
Classification Schema (10)

Outcome ~ Column 1 + Column 2 + Column 3

<div>Training Data (50-50000 rows)</div> <div>Test Data (50-5000 rows)</div>		<div>Modeling Data (100-50000 rows)</div>	Input Column 1	Input Column 2	Input Column 3	Outcome Column
			0.123	red	1	Yes
			0.987	green	1	No
			0.245	blue	0	Yes
			0.254	blue	1	Yes
			0.244	blue	0	No
			0.415	green	0	Maybe
			0.925	red	1	Yes
			0.376	green	0	Yes
			0.615	green	1	No
			0.321	blue	0	Maybe
			0.098	green	0	No
			0.765	red	1	No

Classification Schema (11)

Outcome ~ Column 1 + Column 2 + Column 3



Classification Schema (12)

Outcome ~ Column 1 + Column 2 + Column 3

Model

Input Column 1	Input Column 2	Input Column 3	Outcome Column
0.123	red	1	Yes
0.987	green	1	No
0.245	blue	0	Yes
0.254	blue	1	Yes
0.244	blue	0	No
0.415	green	0	Maybe
0.925	red	1	Yes
0.376	green	0	Yes
0.615	green	1	No
0.321	blue	0	Maybe
0.098	green	0	No
0.765	red	1	No
0.234	green	1	
0.567	blue	0	
0.890	green	1	
0.314	red	1	
0.310	blue	1	
0.284	blue	1	

Classification Schema (13)

Outcome ~ Column 1 + Column 2 + Column 3

Model

		Input Column 1	Input Column 2	Input Column 3	Outcome Column
Training Data (50-50000 rows)	Modeling Data (100-50000 rows)	0.123	red	1	Yes
		0.987	green	1	No
		0.245	blue	0	Yes
		0.254	blue	1	Yes
		0.244	blue	0	No
0.415		green	0	Maybe	
0.925		red	1	Yes	
0.376		green	0	Yes	
0.615		green	1	No	
0.321		blue	0	Maybe	
0.098		green	0	No	
0.765		red	1	No	
Incremental Data (1 to ? rows)		0.234	green	1	
		0.567	blue	0	
		0.890	green	1	
	0.314	red	1		
	0.310	blue	1		
	0.284	blue	1		

Classification Schema (14)

Outcome ~ Column 1 + Column 2 + Column 3

Model

Incremental Data (1 to ? rows)	Input Column 1	Input Column 2	Input Column 3	Outcome Column
	0.234	green	1	
	0.567	blue	0	
	0.890	green	1	
	0.314	red	1	
	0.310	blue	1	
	0.284	blue	1	

Classification Schema (15)

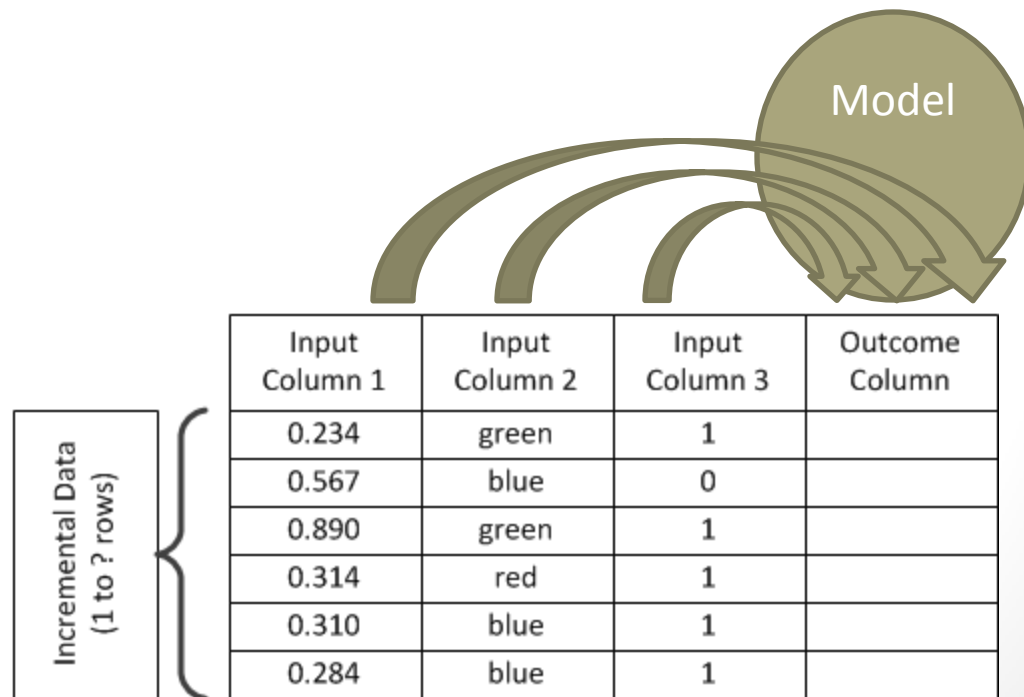
Outcome ~ Column 1 + Column 2 + Column 3

Model

Incremental Data (1 to ? rows)	Input Column 1	Input Column 2	Input Column 3	Outcome Column
	0.234	green	1	
	0.567	blue	0	
	0.890	green	1	
	0.314	red	1	
	0.310	blue	1	
	0.284	blue	1	

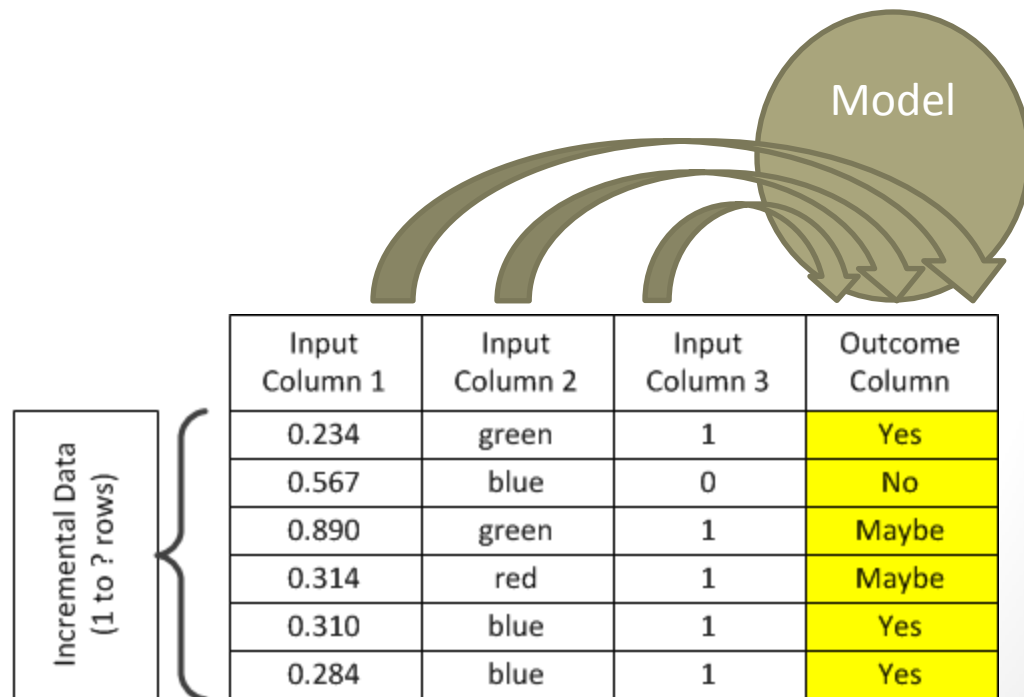
Classification Schema (16)

Outcome \sim Column 1 + Column 2 + Column 3



Classification Schema (17)

Outcome ~ Column 1 + Column 2 + Column 3



Classification Schema (18)

Outcome ~ Column 1 + Column 2 + Column 3

Model

Incremental Data (1 to ? rows)	Input Column 1	Input Column 2	Input Column 3	Outcome Column
	0.234	green	1	Yes
	0.567	blue	0	No
	0.890	green	1	Maybe
	0.314	red	1	Maybe
	0.310	blue	1	Yes
	0.284	blue	1	Yes

Classification Schema (19)

- Attributes
 - All the columns are attributes
- Input Column
 - Input columns are columns that can help predict the outcome. Input columns can be of type binary, ordinal, or category.
- Target Outcome
 - The term "Target Outcome" is redundant. The outcome is the target and vice versa. The target or outcome is the output of a predict function. Providing target or outcome values during modeling makes the process supervised. Creating a model using a outcome is called supervised learning.
- Proxy Column
 - A proxy column is a column that predicts too well. It is too good to be true. Something from the target leaked. This is also called target leakage. The leaked information is "not fair" to use in a prediction. Values for that attribute will not be available when you want to predict the target outcome.
- Key Column
 - In principle, a key column should not affect the model's prediction. The relationship between a key and any other attribute should be random. In practice, the algorithm will find a pattern in the key column and train on this pattern. This pattern is likely to be fortuitous, that means: random. The pattern will not hold for test data or when the model is applied. As a consequence, the key column will affect the model in a bad way.
- Constant Column
 - A constant column should have no affect on the model's predictions. The constant column may increase computation time and cause other problems. It is standard practice to remove all constant columns prior to modeling.

Classification Schema

Break

Partition Modeling Data

How to Partition Data (0)

- The topic of these slides are data partition but we will also perform classifications in R.
- Open in R studio:
 - ClassificationInR.R
 - ClassificationHelper.R
- Run ClassificationInR.R (source)

How to Partition Data (1)

- Test data need to be derived from the same data source as the training data
- Partition of Data between Test and Training must be random
- For an example, open in R Studio:
 - ClassificationInR.R
 - ClassificationHelper.R
 - Find the function definition of BadPartition() in ClassificationHelper.R
 - See how to use the function BadPartition() in ClassificationInR.R

How to Partition Data (2)

The Wrong Way

1. Specify test fraction (e.g. split off 30% or 40% for testing)
2. Take the first fraction of the data as test data
3. Take the rest of the data as training data

How to Partition Data (3)

The Wrong Way

1. Get the DATAFRAME and the testFraction (testFraction default is 0.3)
2. `numberOfRows <- nrow(DATAFRAME)`
3. `numberOfTestRows <- testFraction * numberOfRows`
4. `testSelection <- 1:numberOfRows <= numberOfTestRows`
5. `testData <- DATAFRAME[testSelection,]`
6. `trainData <- DATAFRAME[!testSelection,]`

How to Partition Data (4)

Slow and Clean

1. Specify test fraction (e.g. split off 30% or 40% for testing)
2. Generate random number for each case
3. Create Flag to partition cases: find quantile
 - a) Sort random numbers
 - b) Determine threshold where quantile is the test fraction
 - c) Compare random numbers to threshold.
4. Apply Flag for Test Data selection
5. Apply Flag for Train Data selection

How to Partition Data (5)

The Exact Way

1. Get the DATAFRAME and the testFraction (testFraction default is 0.3)
2. `randoms <- runif(nrow(DATAFRAME))`
3. # Create Test Selection
 - a) `sortedRandoms <- sort(randoms)`
 - b) `testThreshold <- sortedRandoms[length(sortedRandoms)*testFraction]`
 - c) `testSelection <- randoms <= testThreshold`
4. `testData <- DATAFRAME[testSelection,]`
5. `trainData <- DATAFRAME[!testSelection,]`

How to Partition Data (6)

The Fast Way

1. Specify test fraction (e.g. split off 30% or 40% for testing)
2. Generate random number for each case
3. Create Flag to partition cases: test fraction
4. Apply Flag for Test Data selection
5. Apply Flag for Train Data selection

How to Partition Data (7)

The Fast Way

1. Get the DATAFRAME and the testFraction (testFraction default is 0.3)
2. `randoms <- runif(nrow(DATAFRAME))`
3. # Create Test Selection
 - a) `testSelection <- randoms <= testFraction`
4. `testData <- DATAFRAME[testSelection,]`
5. `trainData <- DATAFRAME[!testSelection,]`

How to Partition Data (8)

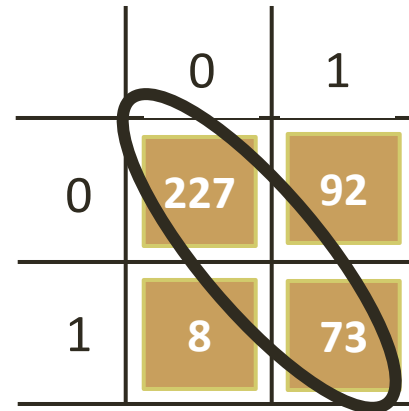
Take Home Message

1. For supervised learning, partition the modeling data in a test set and a train set
2. Use the test data to check the performance of an algorithm
3. A confusion matrix and an ROC chart can be used to test classifications
4. Classification accuracy can be defined as the number of correct predictions divided by the number of predictions or test cases

How to Partition Data (9)

Take Home Messages

1. For supervised learning, partition the modeling data in a test set and a train set
2. Use the test data to check the performance of an algorithm
3. A confusion matrix and an ROC chart can be used to test classifications
4. Classification accuracy can be defined as the number of correct predictions divided by the number of predictions or test cases



	0	1
0	227	92
1	8	73

How to Partition Data (10)

Take Home Messages

1. For supervised learning, partition the modeling data in a test set and a train set
2. Use the test data to check the performance of an algorithm
3. A confusion matrix and an ROC chart can be used to test classifications
4. Classification accuracy can be defined as the number of correct predictions divided by the number of predictions or test cases

	0	1
0	227	92
1	8	73

$$75\% = \left(227 + 73 \right) \div \left(8 + 92 + 227 + 73 \right)$$

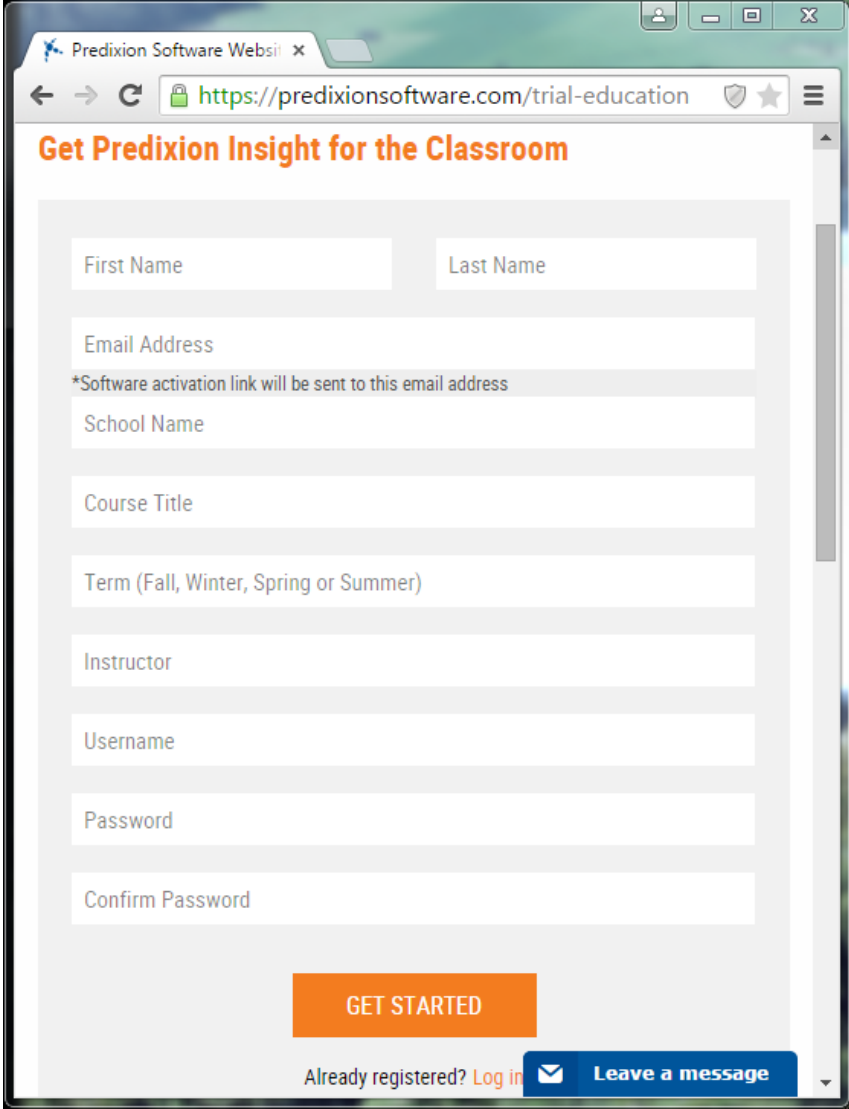
Partition Modeling Data

Predixion Insight

Predixion Insight

- You will get a free educational license for Predixion Insight
- Go to this address (Please do not distribute this address):

<http://predixionsoftware.com/trial-education>



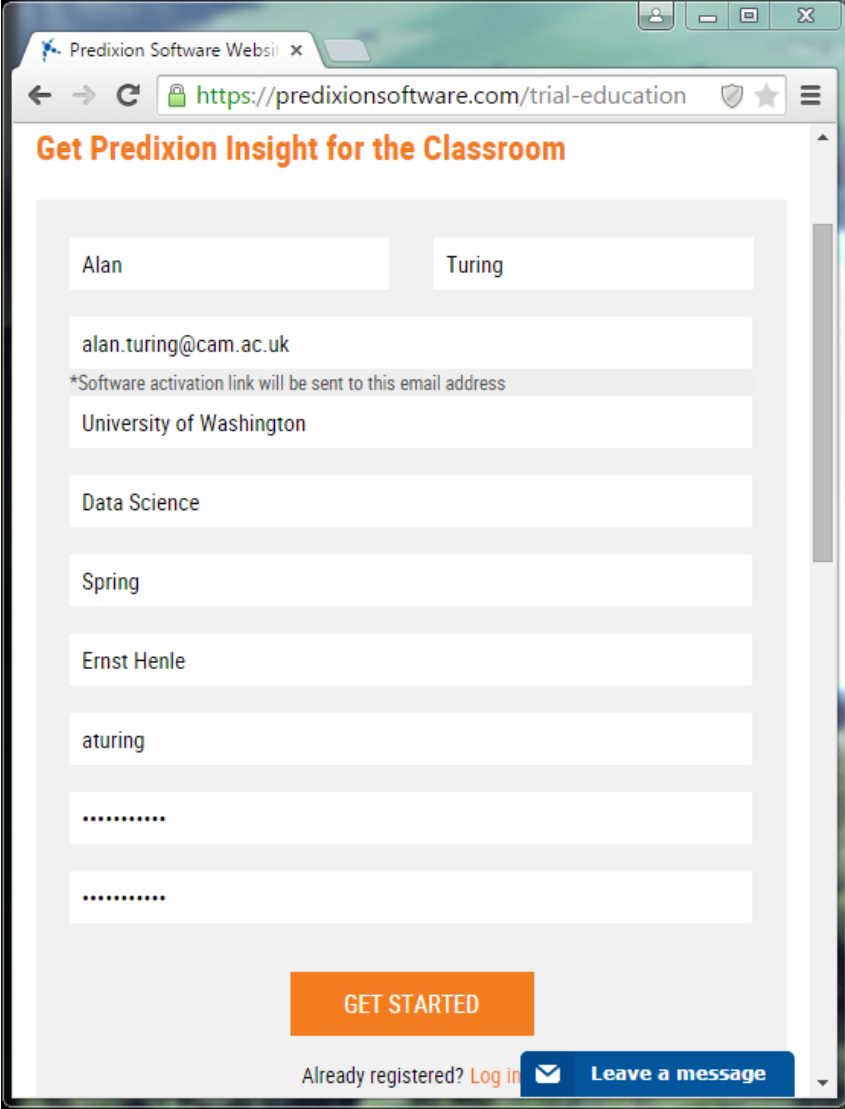
The screenshot shows a web browser window with the address bar displaying <https://predixionsoftware.com/trial-education>. The page title is "Predixion Software Website". The main heading is "Get Predixion Insight for the Classroom". Below the heading is a registration form with the following fields:

- First Name
- Last Name
- Email Address
- *Software activation link will be sent to this email address
- School Name
- Course Title
- Term (Fall, Winter, Spring or Summer)
- Instructor
- Username
- Password
- Confirm Password

At the bottom of the form is an orange button labeled "GET STARTED". Below the button, there is a link "Already registered? Log in" and a blue button with a white envelope icon labeled "Leave a message".

Predixion Insight

- Fill in the requested information
- Submit the request by pressing the “GET STARTED” button.
- Go to your email and click the GET STARTED button.
- You will get a response:
 - **Your trial has not been activated yet.** An email has been sent to:
[alan.turing@cam.ac.uk.com](mailto:alan.turing@cam.ac.uk)
 - Please click the activation link in the email



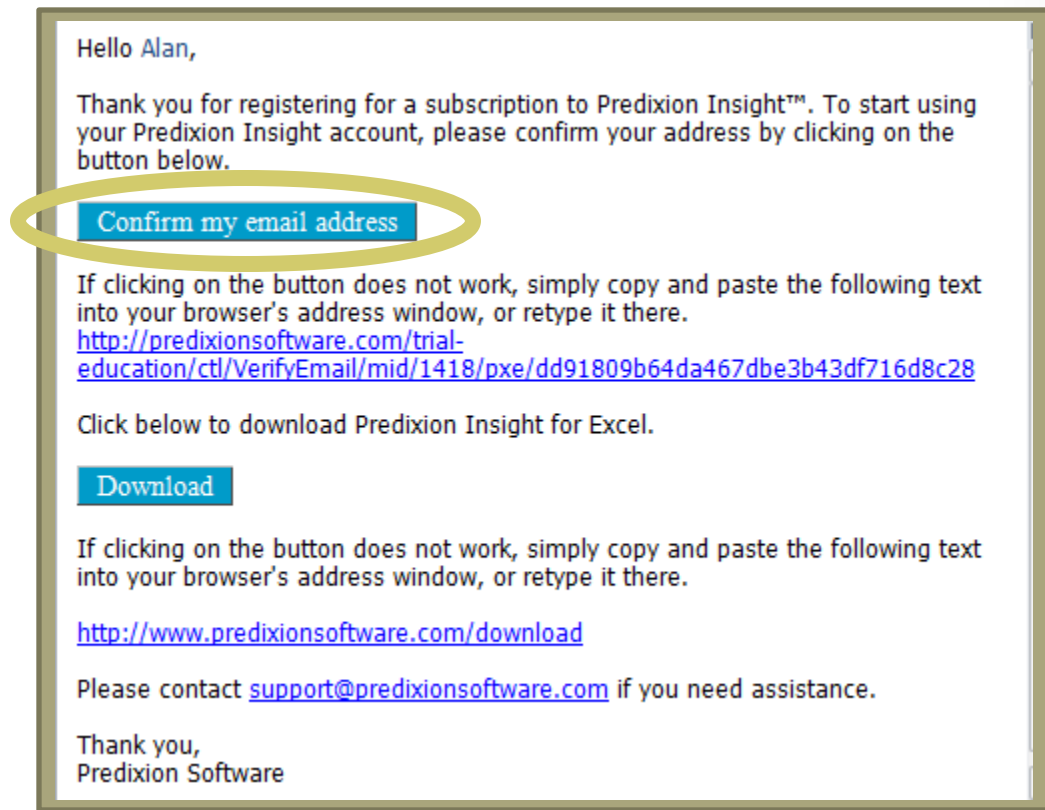
The screenshot shows a web browser window with the address bar displaying <https://predixionsoftware.com/trial-education>. The page title is "Predixion Software Website". The main heading is "Get Predixion Insight for the Classroom". The form contains the following fields and text:

- First Name: Alan
- Last Name: Turing
- Email: alan.turing@cam.ac.uk
- Text below email: *Software activation link will be sent to this email address
- Institution: University of Washington
- Field: Data Science
- Field: Spring
- Field: Ernst Henle
- Field: aturing
- Field:
- Field:

At the bottom right is an orange "GET STARTED" button. Below the button, it says "Already registered? Log in" followed by a blue button with a white envelope icon and the text "Leave a message".

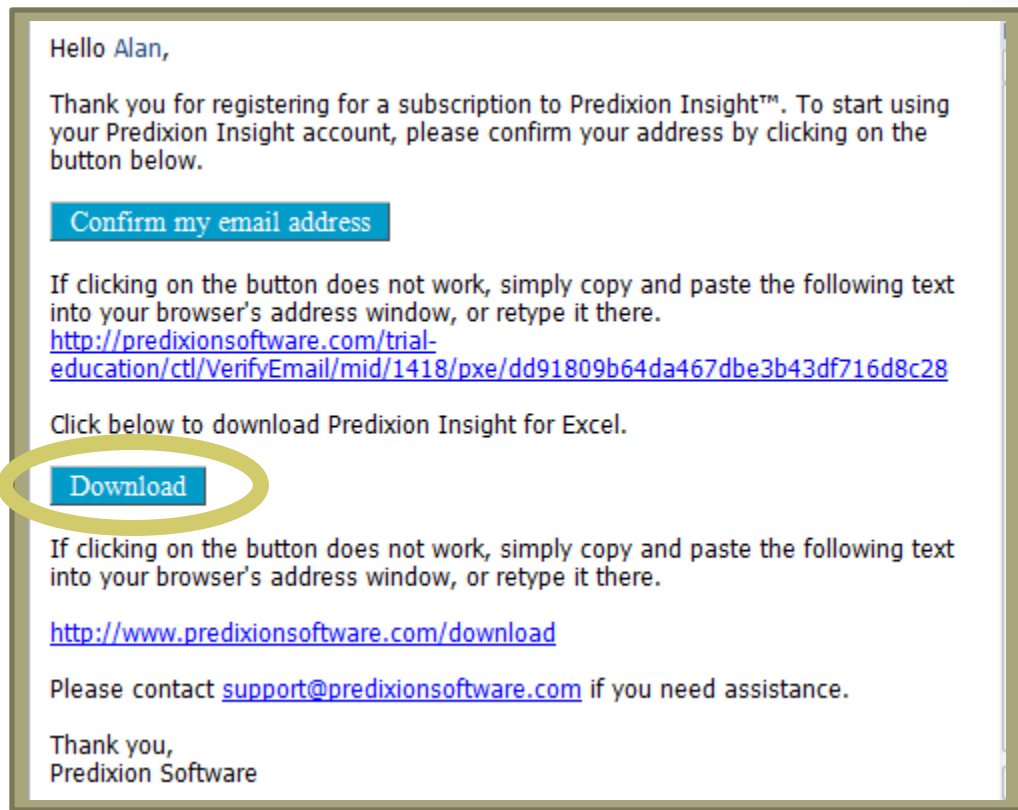
Predixion Insight

- Go to your email. You will see an email from support@predixionsoftware.com
- In that email click on the button “Confirm my email address”



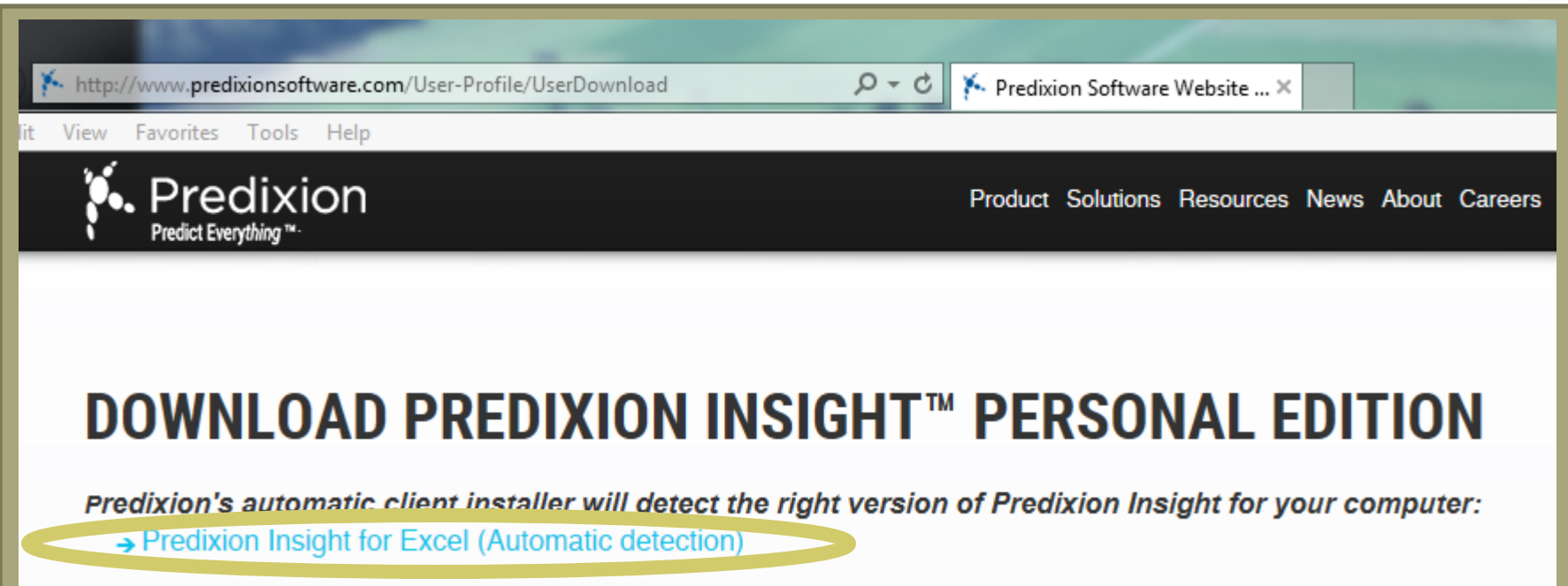
Predixion Insight

- Predixion Insight server is in the cloud but the Predixion Insight client is an Excel add-in. You will need Excel 2010 or later running on Windows 7 or later. Please contact me if these requirements pose a problem.
- In the same email as before, click “Download” to download the Predixion Insight add-in for Excel.



Predixion Insight

- Click on Predixion Insight for Excel (Automatic detection)
- Run the PXClientInstallLauncher.exe and go through the wizard



The screenshot shows a web browser window with the address bar displaying <http://www.predixionsoftware.com/User-Profile/UserDownload>. The browser's menu bar includes "File", "View", "Favorites", "Tools", and "Help". The website's header features the Predixion logo with the tagline "Predict Everything™" on the left, and navigation links for "Product", "Solutions", "Resources", "News", "About", and "Careers" on the right. The main content area has a large heading "DOWNLOAD PREDIXION INSIGHT™ PERSONAL EDITION". Below this, a message states: "Predixion's automatic client installer will detect the right version of Predixion Insight for your computer:". A blue hyperlink, "→ Predixion Insight for Excel (Automatic detection)", is highlighted with a yellow oval.

<http://www.predixionsoftware.com/User-Profile/UserDownload>

Predixion
Predict Everything™

Product Solutions Resources News About Careers

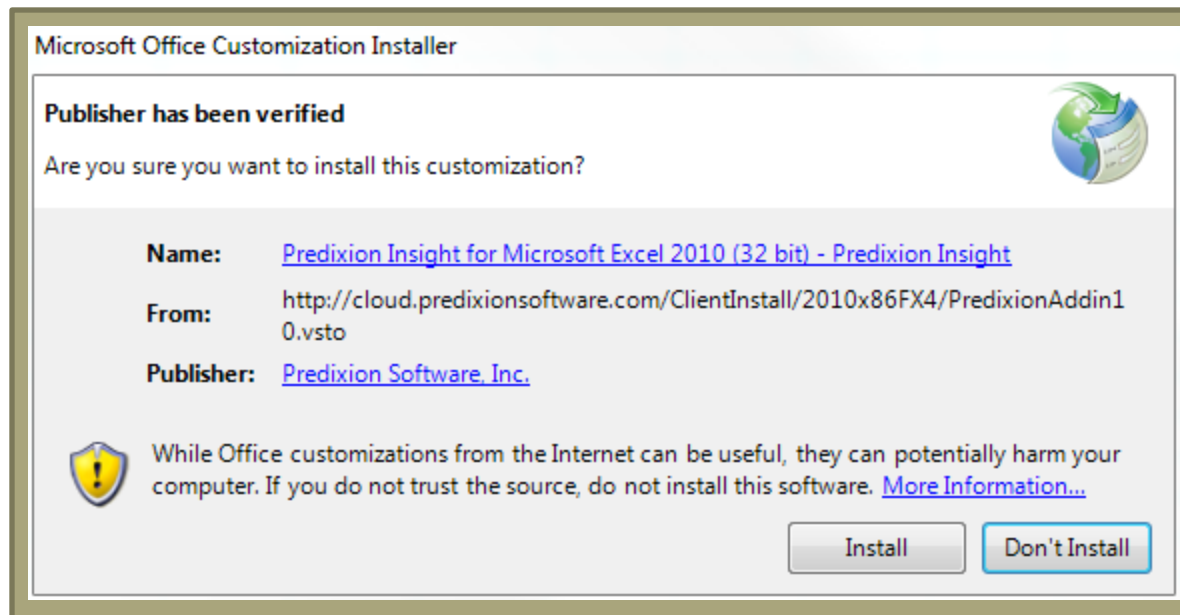
DOWNLOAD PREDIXION INSIGHT™ PERSONAL EDITION

Predixion's automatic client installer will detect the right version of Predixion Insight for your computer:

[→ Predixion Insight for Excel \(Automatic detection\)](#)

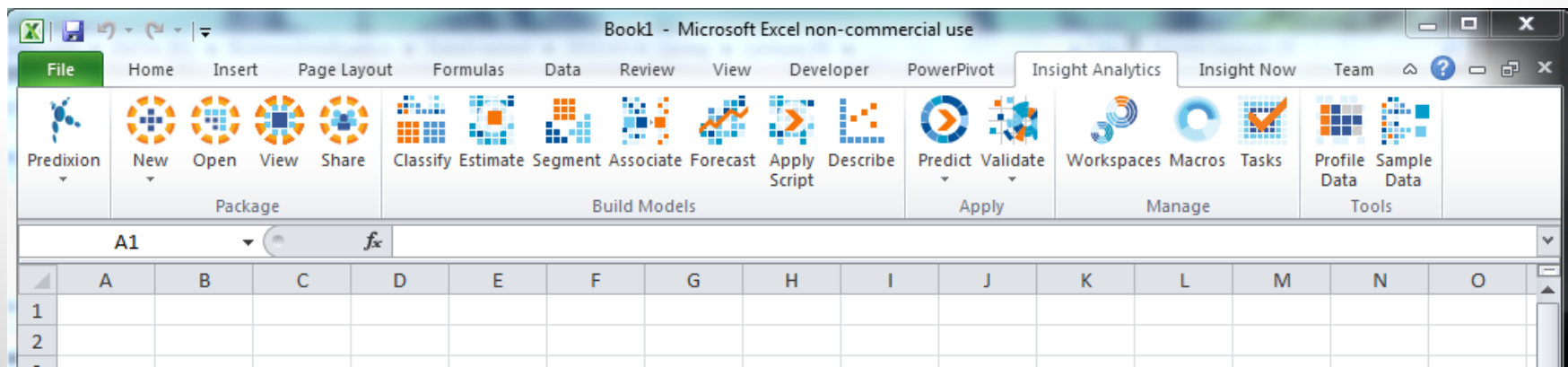
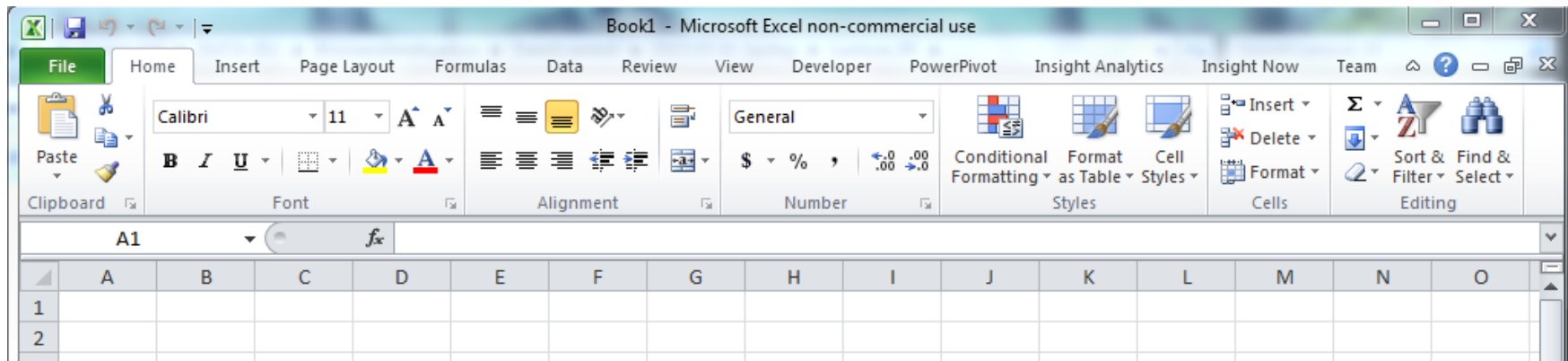
Predixion Insight

- Excel will start on its own. Or you should start Excel.
- The Microsoft Office Customization Installer will appear. Click Install.



Predixion Insight

- Excel will appear with two additional ribbons
- Click on the Insight Analytics ribbon



Predixion Insight

- Do the following walkthrough only up to and including Step 5 “Creating a Classification Model - Test Models”:
https://www.predixionsoftware.com/help/webframe.html#01_ClassifyWalkthrough.html. Contact me if you need help with the walkthrough or with a windows machine that has Excel 2010 (or later). Make a screen shot of a side-by-side ROC chart and classification / confusion matrix .

Get a Predixion Insight License

Setup Virtual Machine

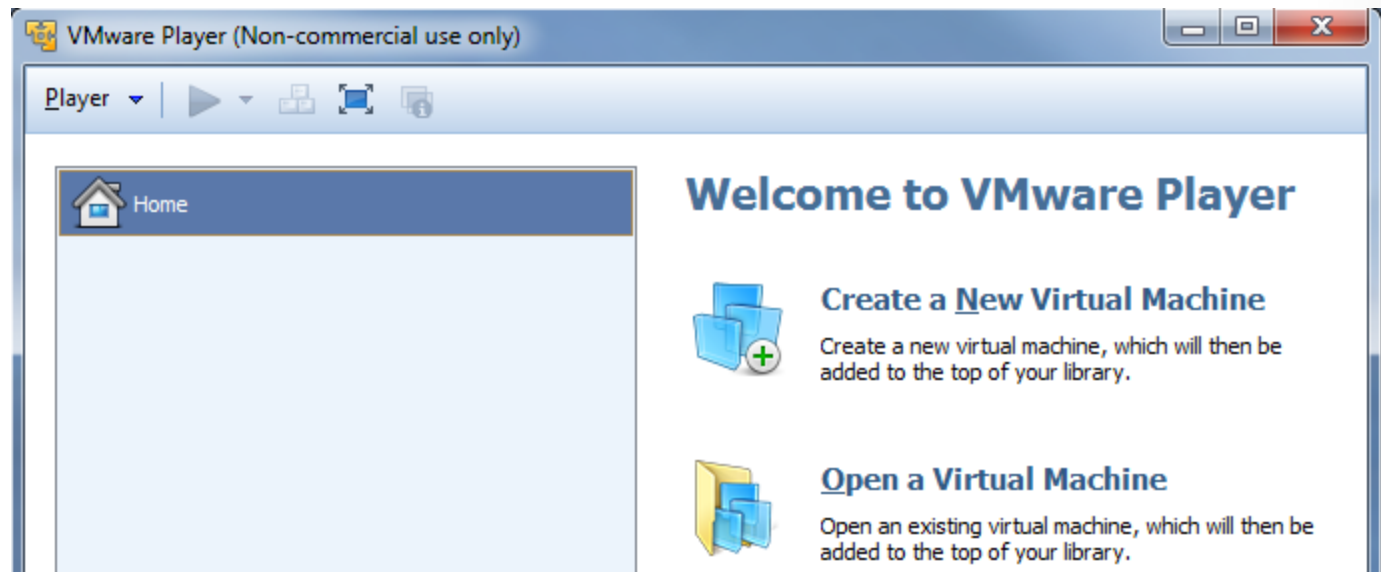
Setup VM (0)

- Virtual Box may be a better alternative than VMWare for Mac users
- To download and install Virtual Box, follow the instructions in this website: <https://www.virtualbox.org/wiki/Downloads>
- In particular, for a Mac OS X, use the link for: VirtualBox 4.3.20 for OS X hosts x86/amd64



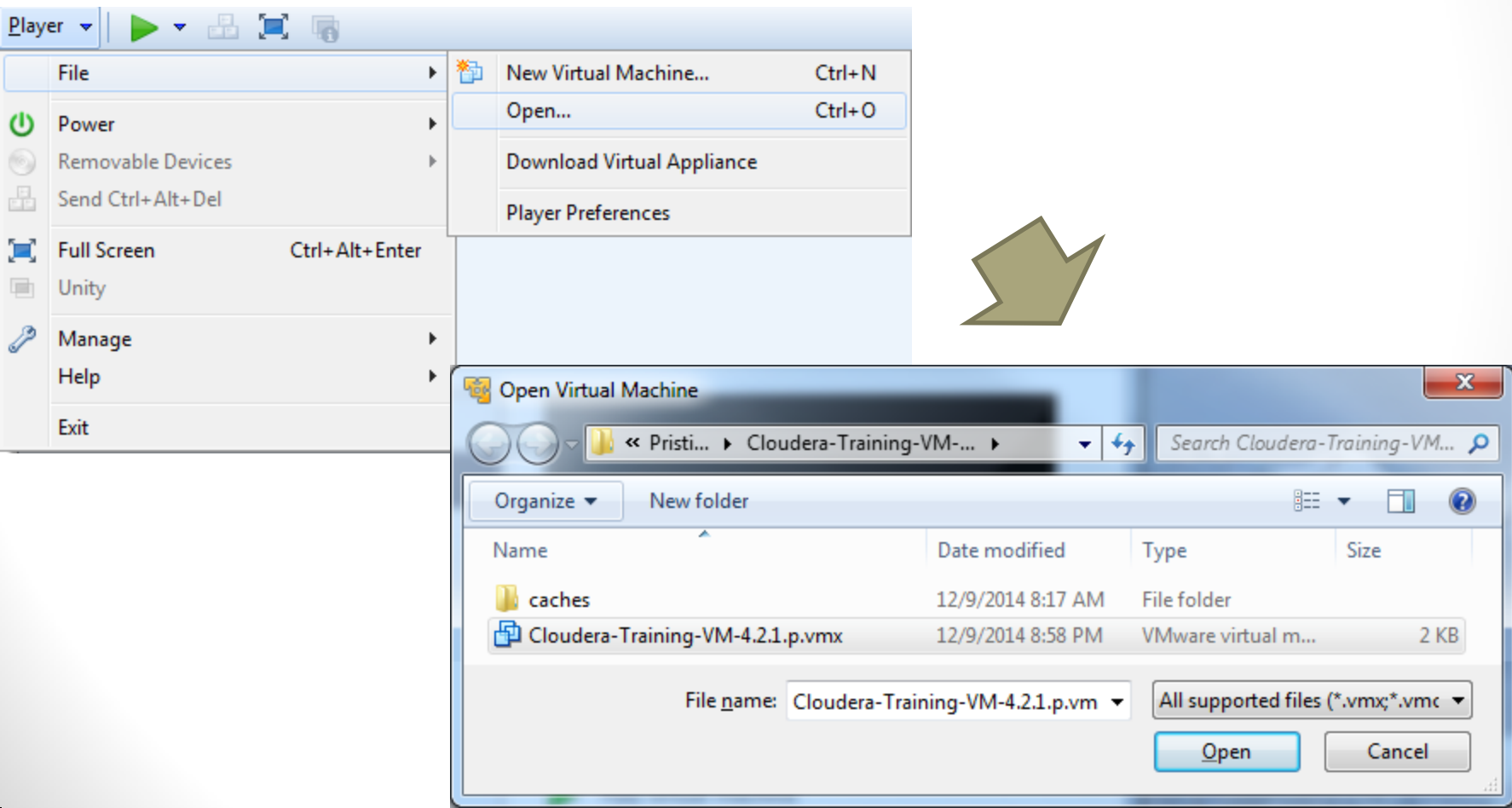
Setup VM (1)

- Download VMWare Player (Windows) or VMWare Fusion (Mac)
- Unzip Cloudera-Training-VM-4.2.1.p-vmware_prist.zip
- Start VMWare or Virtual Box



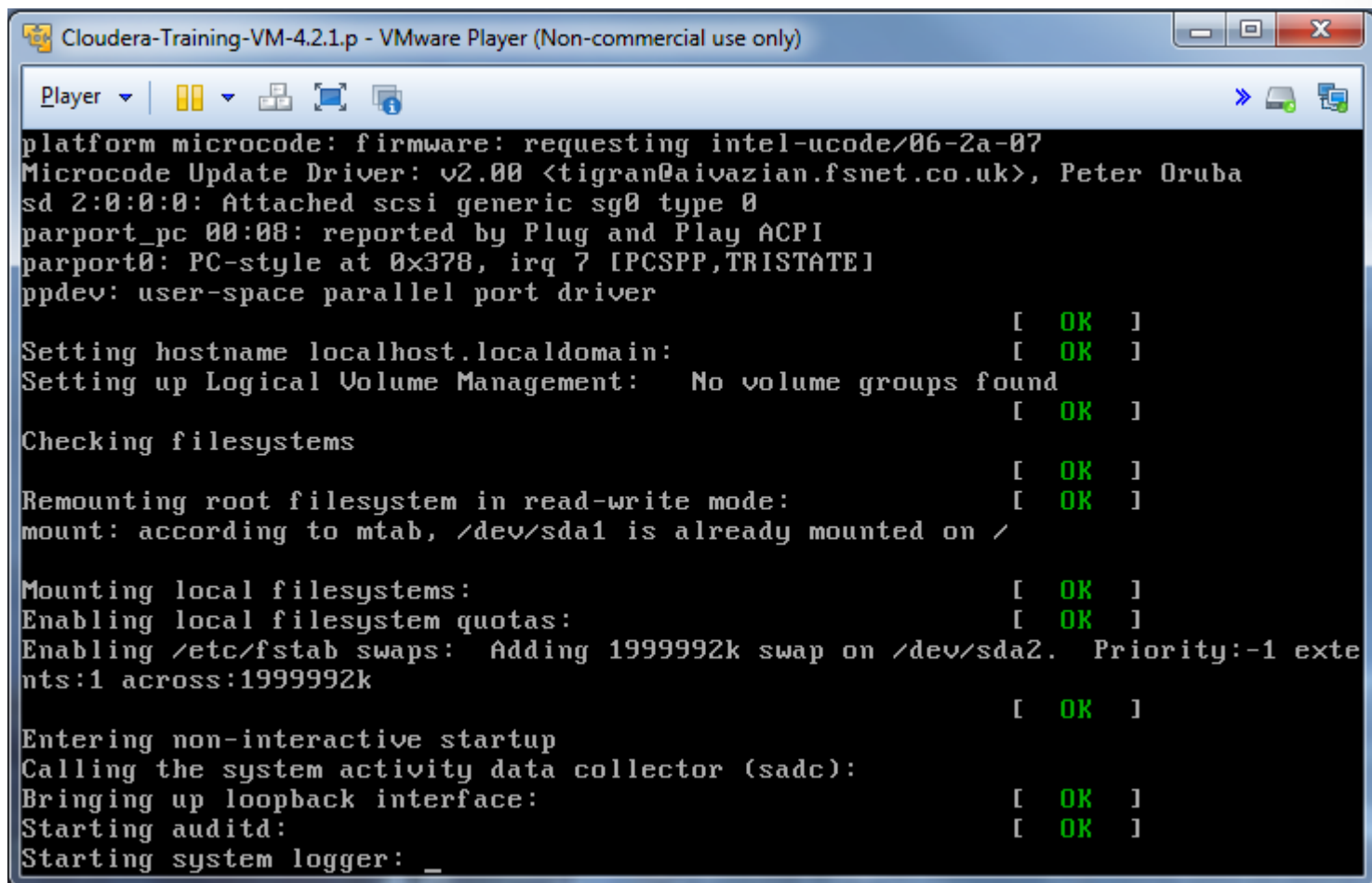
Setup VM (2)

- Open the VM: Select Cloudera-Training-VM-4.2.1.p.vmx



Setup VM (3)

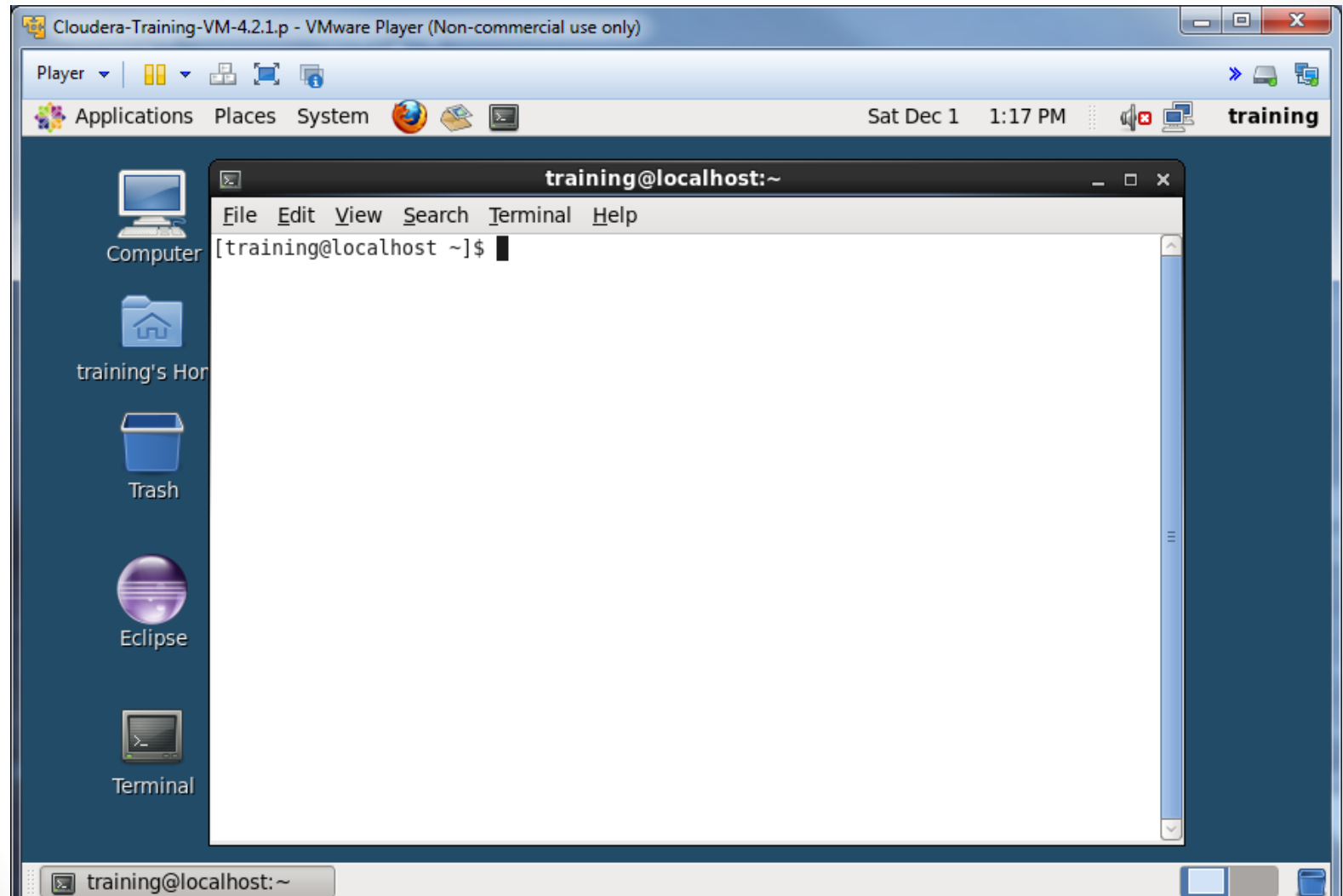
- Note: The VM starts up



```
Cloudera-Training-VM-4.2.1.p - VMware Player (Non-commercial use only)
Player | [Icons]
platform microcode: firmware: requesting intel-ucode/06-2a-07
Microcode Update Driver: v2.00 <tigran@aivazian.fsnet.co.uk>, Peter Oruba
sd 2:0:0:0: Attached scsi generic sg0 type 0
parport_pc 00:08: reported by Plug and Play ACPI
parport0: PC-style at 0x378, irq 7 [PCSP,TRISTATE]
ppdev: user-space parallel port driver
[ OK ]
Setting hostname localhost.localdomain: [ OK ]
Setting up Logical Volume Management: No volume groups found
[ OK ]
Checking filesystems
[ OK ]
Remounting root filesystem in read-write mode: [ OK ]
mount: according to mtab, /dev/sda1 is already mounted on /
Mounting local filesystems: [ OK ]
Enabling local filesystem quotas: [ OK ]
Enabling /etc/fstab swaps: Adding 1999992k swap on /dev/sda2. Priority:-1 exte
nts:1 across:1999992k
[ OK ]
Entering non-interactive startup
Calling the system activity data collector (sadc):
Bringing up loopback interface: [ OK ]
Starting auditd: [ OK ]
Starting system logger: _
```

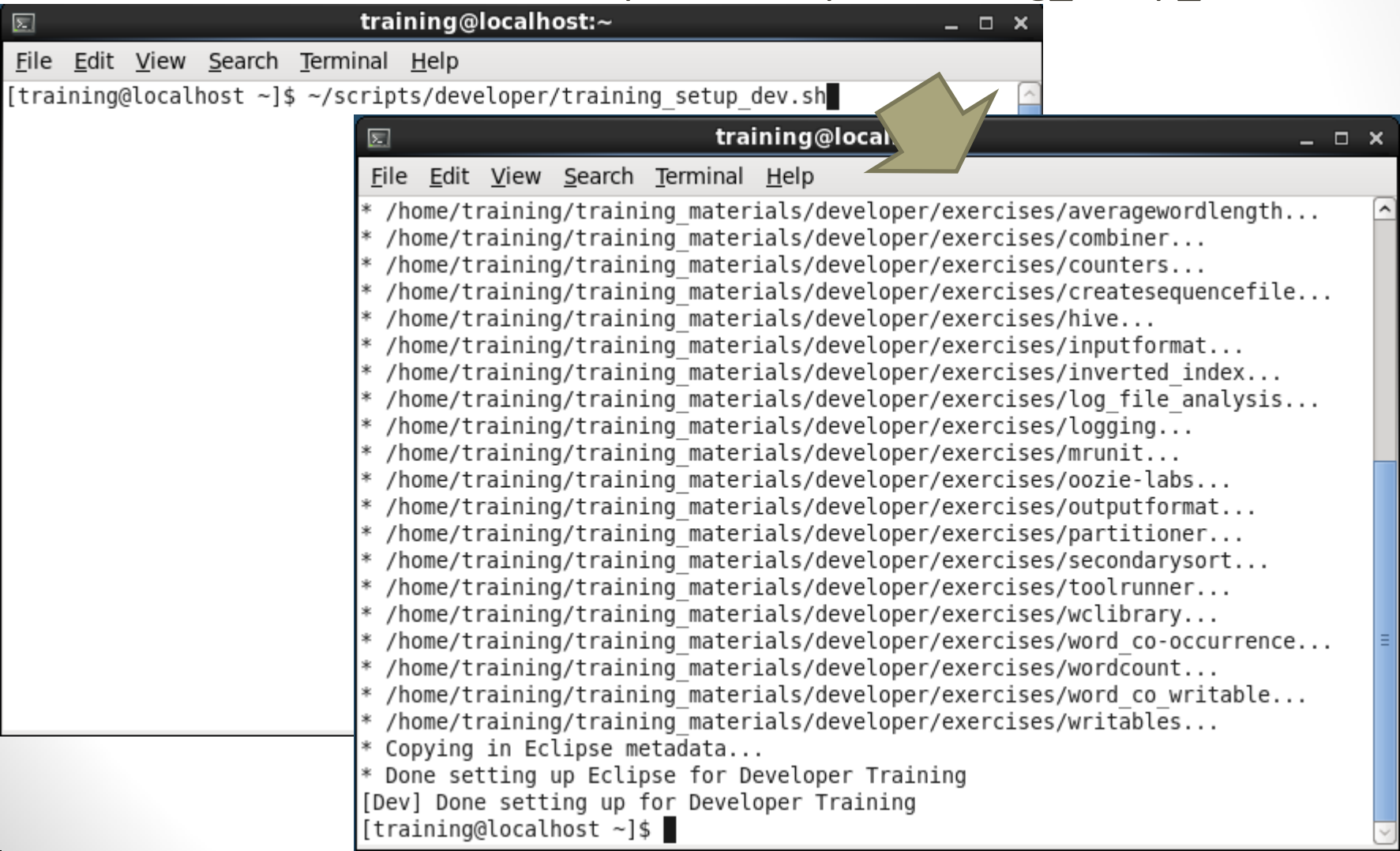
Setup VM (4)

- Note: The VM is running



Setup VM (5)

- Enter into Console: `~/scripts/developer/training_setup_dev.sh`



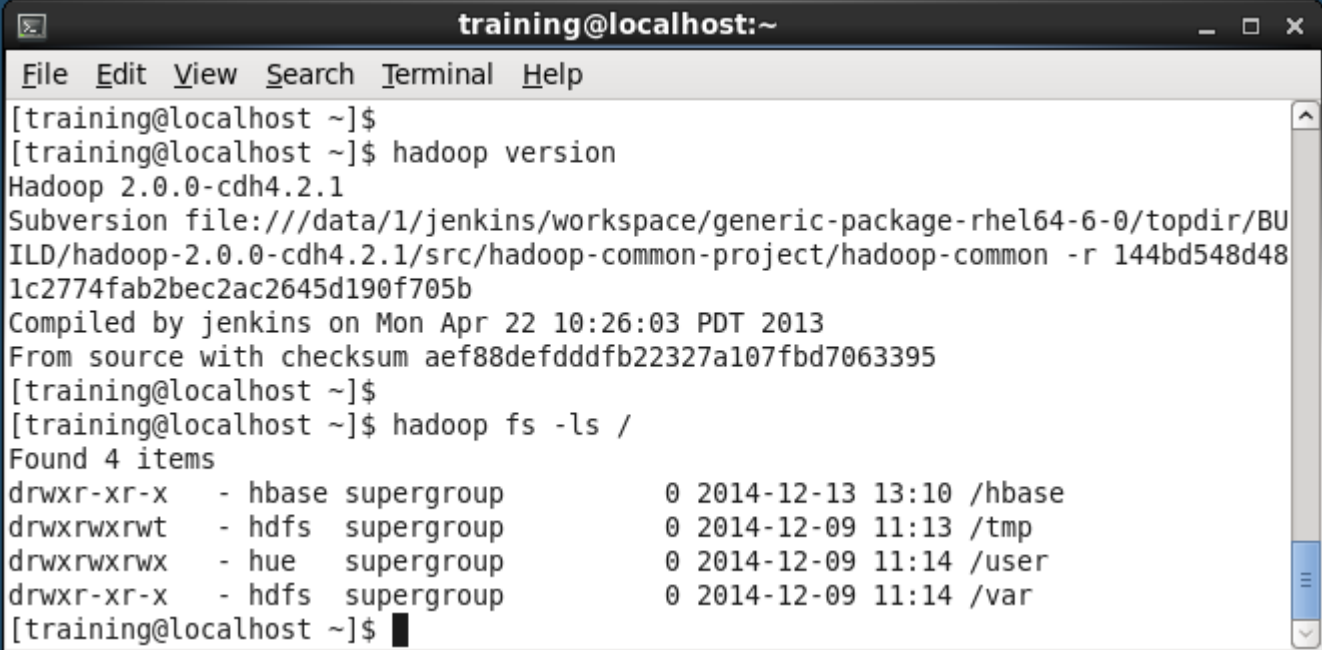
```
training@localhost:~  
File Edit View Search Terminal Help  
[training@localhost ~]$ ~/scripts/developer/training_setup_dev.sh
```



```
training@localhost:~  
File Edit View Search Terminal Help  
* /home/training/training_materials/developer/exercises/averagewordlength...  
* /home/training/training_materials/developer/exercises/combiner...  
* /home/training/training_materials/developer/exercises/counters...  
* /home/training/training_materials/developer/exercises/createsequencefile...  
* /home/training/training_materials/developer/exercises/hive...  
* /home/training/training_materials/developer/exercises/inputformat...  
* /home/training/training_materials/developer/exercises/inverted_index...  
* /home/training/training_materials/developer/exercises/log_file_analysis...  
* /home/training/training_materials/developer/exercises/logging...  
* /home/training/training_materials/developer/exercises/mrunit...  
* /home/training/training_materials/developer/exercises/oozie-labs...  
* /home/training/training_materials/developer/exercises/outputformat...  
* /home/training/training_materials/developer/exercises/partitioner...  
* /home/training/training_materials/developer/exercises/secondarysort...  
* /home/training/training_materials/developer/exercises/toolrunner...  
* /home/training/training_materials/developer/exercises/wclibrary...  
* /home/training/training_materials/developer/exercises/word_co-occurrence...  
* /home/training/training_materials/developer/exercises/wordcount...  
* /home/training/training_materials/developer/exercises/word_co_writable...  
* /home/training/training_materials/developer/exercises/writables...  
* Copying in Eclipse metadata...  
* Done setting up Eclipse for Developer Training  
[Dev] Done setting up for Developer Training  
[training@localhost ~]$
```

Setup VM (6)

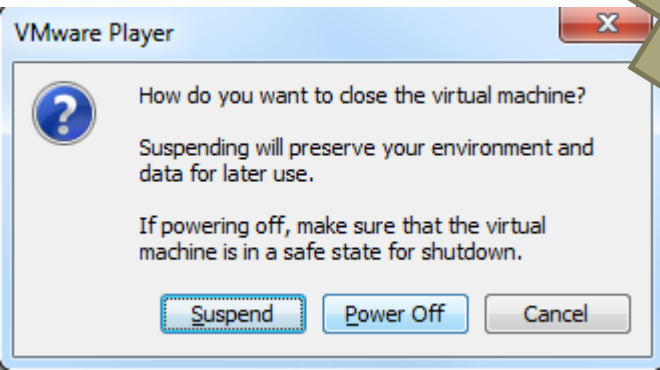
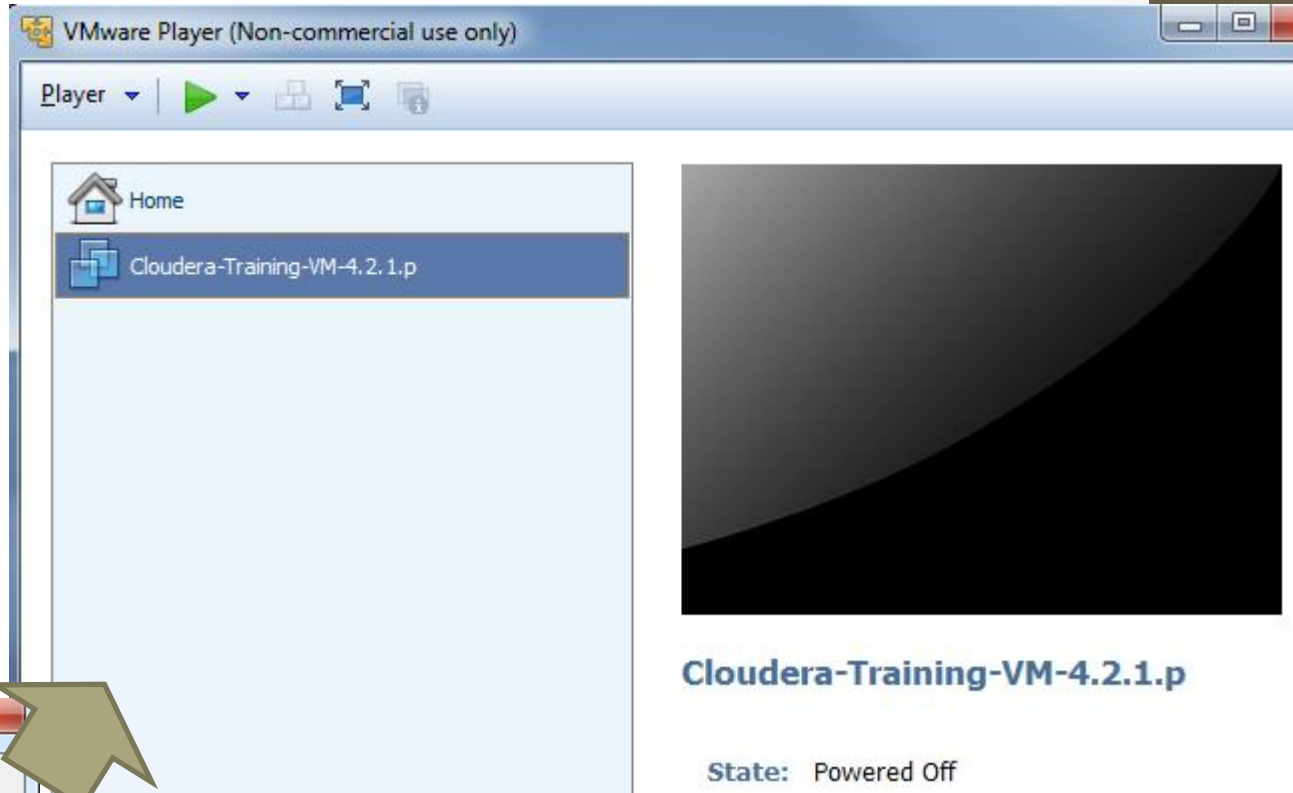
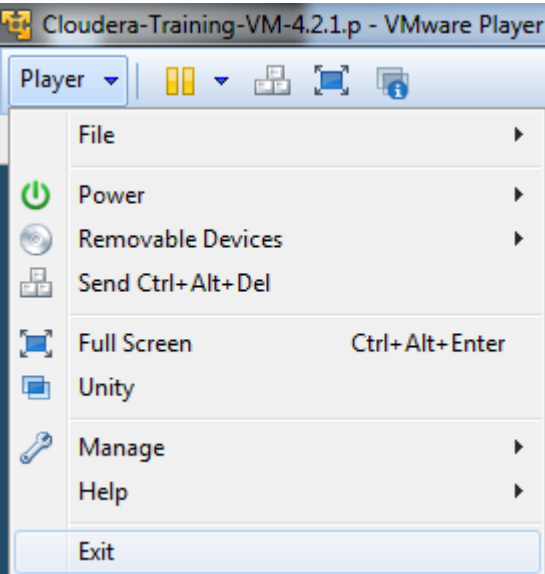
- Enter into Console: `hadoop version`
- Enter into Console: `hadoop fs -ls /`



```
training@localhost:~  
File Edit View Search Terminal Help  
[training@localhost ~]$  
[training@localhost ~]$ hadoop version  
Hadoop 2.0.0-cdh4.2.1  
Subversion file:///data/1/jenkins/workspace/generic-package-rhel64-6-0/topdir/BU  
ILD/hadoop-2.0.0-cdh4.2.1/src/hadoop-common-project/hadoop-common -r 144bd548d48  
1c2774fab2bec2ac2645d190f705b  
Compiled by jenkins on Mon Apr 22 10:26:03 PDT 2013  
From source with checksum aef88defdddfb22327a107fbd7063395  
[training@localhost ~]$  
[training@localhost ~]$ hadoop fs -ls /  
Found 4 items  
drwxr-xr-x - hbase supergroup          0 2014-12-13 13:10 /hbase  
drwxrwxrwt - hdfs supergroup           0 2014-12-09 11:13 /tmp  
drwxrwxrwx - hue supergroup            0 2014-12-09 11:14 /user  
drwxr-xr-x - hdfs supergroup           0 2014-12-09 11:14 /var  
[training@localhost ~]$
```

Setup VM (7)

- Shutdown VM (select Exit and Power Off)



Setup VM (8)

- Optional: Increase memory if you have enough RAM to spare.
Set to 2 GB.

Cloudera-Training-VM-4.2.1.p


State: Powered Off

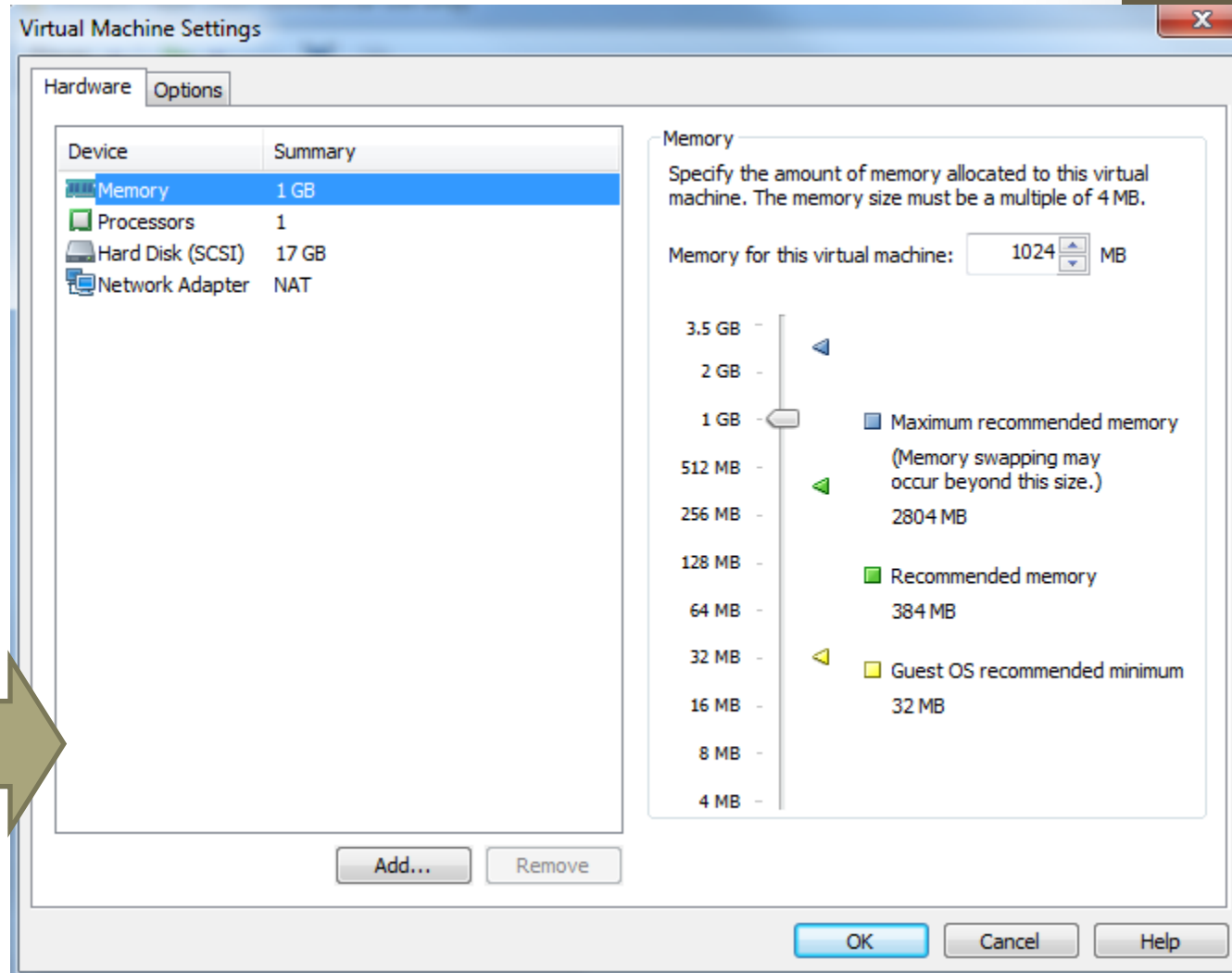
OS: Other Linux 64-bit

Version: Workstation 5.x virtual machine

RAM: 1 GB

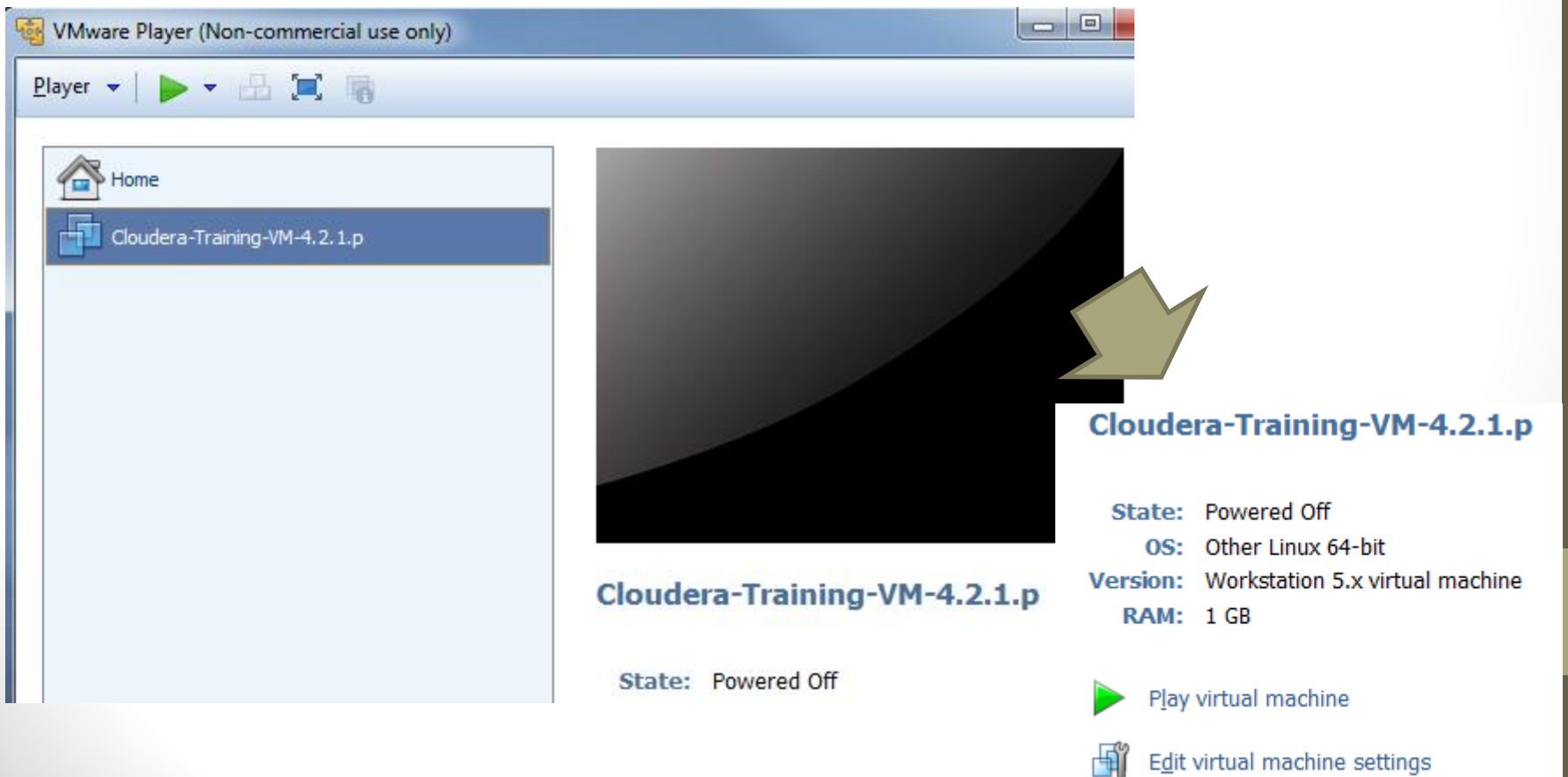
 Play virtual machine

 Edit virtual machine settings



Setup VM (9)

- To start up VM again:
 - Select Virtual Machine (Cloudera-Training-VM-4.2.1.p)
 - Play Virtual Machine



Setup Virtual Machine

Review: Terminology

- Algorithm
- Anomaly detection
- Association
- Attribute
- Binarize Categories
- Binary Column
- Case
- Category Column
- Character Column
- Classification
- Clustering
- Coercion
- Column
- Column Header
- Data
- Data Dimensionality
- Data Frame
- Data Type
- DFD
- Dummy Variable
- Estimation
- Feature Scaling
- Field
- Hypothesis
- Key Column
- Machine Learning
- Market-basket analysis
- MATLAB
- Matrix
- Missing Data
- Model
- Multinomial Column
- Normalization
- Numeric Column
- Observation
- Outcome
- Outlier Removal
- Predictive Analytics
- R
- Rectangular Data
- Relabeling
- Row
- Schema
- Shaping Data
- Sparse Multi-Dimensional Matrix
- Standard Deviation
- States
- String
- Supervised Learning
- Support
- Table
- Target Column
- Text Column
- Theory
- Un-structured Data
- Unsupervised Learning
- Z-score

Assignment (1)

1. Partition

- a) Add two functions to ClassificationHelper.R. The names of the two functions are: FastPartition and ExactPartition. The functions take in a dataframe and a fraction. The functions return a list of two dataframes. The names of the two dataframes are trainSet and testSet. trainSet and testSet are mutually exclusive cases from the input dataframe. testSet contains the fraction of cases as specified by the fraction input. trainSet contains the rest. The assignments of cases to testSet or trainSet are random.
- b) Run the script (source): ClassificationInR.R using BadPartition()
- c) Note the confusion matrix
- d) In ClassificationInR.R replace the line containing BadPartition() with FastPartition()
- e) Run the script (source): ClassificationInR.R
- f) Note how the confusion matrix changed

2. Classification in R

- a) In ClassificationInR.R add code to create a Naive Bayes Model. You will have to look up the Naive Bayes package "e1071" to determine the inputs. Use the formula in ClassificationInR.R. Get help from the LinkedIn group.
- b) In ClassificationInR.R add code to predict outcomes based on the Naive Bayes model. You will have to read the documentation to determine the "type" parameter. **It is very important that you answer for yourself: How many rows are there in the outcome? How many columns? How many columns are in predictedProbabilities.GLM?**
- c) Add code to create a confusion matrix like the one for the logistic regression.
- d) Based on the confusion matrix, what is the accuracy when using BadPartition()? Accuracy is defined as the correct predictions divided by all predictions.
- e) Based on the confusion matrix, what is the accuracy when using ExactPartition()? Accuracy is defined as the correct predictions divided by all predictions.

Assignment (2)

3. Get a Predixion License. See the section of the slides titled “Predixion Insight”. Do the following walkthrough only up to and including Step 5 “Creating a Classification Model - Test Models”: https://www.predixionsoftware.com/help/webframe.html#01_ClassifyWalkthrough.html. Contact me if you need help with the walkthrough or with a windows machine that has Excel 2010 (or later). Make a screen shot of a side-by-side ROC chart and classification / confusion matrix .
4. Unzip the Hadoop VM and run the script as indicated in the slides titled “Setup Virtual Machine” (`~/scripts/developer/training_setup_dev.sh`). Make a screenshot that includes the last lines of the console.
5. Submit the altered ClassificationHelper.R and ClassificationInR.R to Catalyst by 11 PM Saturday. Submit the screenshot of the ROC and confusion matrix to Catalyst by 11 PM Saturday. Submit the screenshot of the console in the Hadoop VM by 11 PM Saturday.

Assignment (3)

6. Reading Assignments

- Review terminology at the end of this slide deck
- Read Quiz06Practice.txt (Quiz questions for next week)
- Classification
 - Read ROC Curve, Lift Chart and Calibration Plot by Vuk and Curk:
<http://mrvar.fdv.uni-lj.si/pub/mz/mz3.1/vuk.pdf>
 - Read about Accuracy:
http://en.wikipedia.org/wiki/Precision_and_recall
 - Read about target leakage:
http://www.cs.umb.edu/~ding/history/470_670_fall_2011/papers/cs670_Tran_PreferedPaper_LeakingInDataMining.pdf
- Relational Model, Relational Algebra, and Relational Calculus
 - http://en.wikipedia.org/wiki/Relational_algebra
 - <http://sentences.com/docs/amd.pdf> (Pages 35 to 48 only)
 - http://en.wikipedia.org/wiki/Relational_model
 - <http://www.youtube.com/watch?v=NvrpuBAMddw>

Introduction to Data Science