

JMP Machine Learning Workshop, London, 19th March 2019

Introduction

The example is taken from the [UCI Machine Learning Repository](#). This is not an example from Science or Engineering. Understanding and mastering the tools of machine learning on this data will empower you to learn more from any difficult data examples that you will come across in your job.


Your Task

You work for 'AdFree' a software company that sells a small application that allows users to browse the web without being bothered by unwanted advertisements.

A key piece of intellectual property is the method used to categorise images as advertisements or non-advertisements, the former being removed from the page before it is displayed. The goal is to build a model that can predict whether a page is an advertisement or not.

You work in a team of analysts trying to improve the current algorithm for ad detection. Working within the team, you can use ANY methods you like to try to minimise the false alarm rate (false positives and false negatives).

The data



UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

[About](#) [Citation Policy](#) [Donate a Data Set](#) [Contact](#)


Search

Repository Web Google

[View ALL Data Sets](#)

Internet Advertisements Data Set

Download: [Data Folder](#), [Data Set Description](#)



Abstract: This dataset represents a set of possible advertisements on Internet pages.

Data Set Characteristics:	Multivariate	Number of Instances:	3279	Area:	Computer
Attribute Characteristics:	Categorical, Integer, Real	Number of Attributes:	1558	Date Donated	1998-07-01
Associated Tasks:	Classification	Missing Values?	Yes	Number of Web Hits:	111513

Source:

Creator & donor:
Nicholas Kushmerick <nick.1@ucd.ie>

Data Set Information:

This dataset represents a set of possible advertisements on Internet pages. The features encode the geometry of the image (if available) as well as phrases occurring in the URL, the image's URL and alt text, the anchor text, and words occurring near the anchor text. The task is to predict whether an image is an advertisement ("ad") or not ("nonad").

Attribute Information:

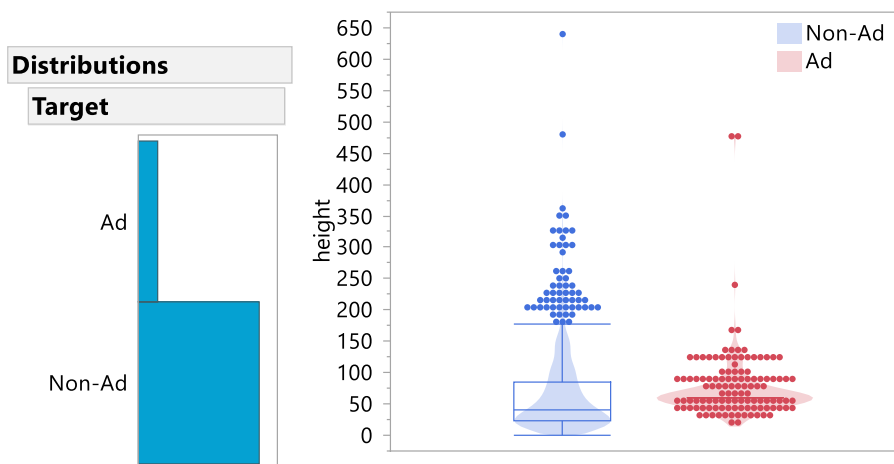
(3 continuous; others binary; this is the "STANDARD encoding" mentioned in the [Kushmerick, 98].)

One or more of the three continuous features are missing in 28% of the instances; missing values should be interpreted as "unknown".

The data are from a variety of web pages. The response is Target and there are 1,558 predictors, including three image geometry measures and binary indicators for words in the URL. There are 2,295 rows. There are many columns. Not all methods will scale well to a data set this size.

The table 'ad_data_training' is available to your team (Add Ins > The 'AdFree' study), and this should be used for building and testing your models until you have identified the one you are most happy with.

In practise you would want to invest time at the start to explore the data using the visual and interactive tools in JMP Pro. This is not the focus of this workshop but you might wish to get more familiar with the data using Distributions or Graph Builder if you are comfortable with these platforms.



Deliverables

You will determine a final, 'best' model and score how well it does by using the data in 'ad_data_scoring'. This table has the same columns as 'ad_data_training' (before you added formula columns to the latter), but has 984 rows. At the end of class, we will give you a password that will let you score your best model and we will see who has the best one. You will be able to see how many rows are mis-classified by your model, and obviously lower scores are better. The model with the fewest mis-classified rows wins a book!

After each module, try some of the methods on the data. Try to come up with the best model possible in the time given.

There is some guidance below to get you started with the important basic functions of each platform. You can go much further than this!

Chris Gotwalt's Tips

The ad data is a large, real data set. Some methods are too slow for this data. You may have to quit JMP if you try a method that is too slow on it. Save your work as you go!

Hands-on session 1: Validation Columns and Neural

The Make Validation Column Utility

1. Select Analyze > Predictive Modeling > Make Validation Column.
(You can also get here another way: Click Validation in a platform launch window)
2. In the Make Validation Column window, you specify the proportion or number of rows for each of your holdback sets. Then you select a method for constructing the holdback sets.
3. Decide on the proportion that you want to use for training and validation (or go with the defaults) and select the method to generate the validation column that you will use for your models.

Chris Gotwalt's Tips

The partitioning of data into training/validation and test sets should reflect the data acquisition process.

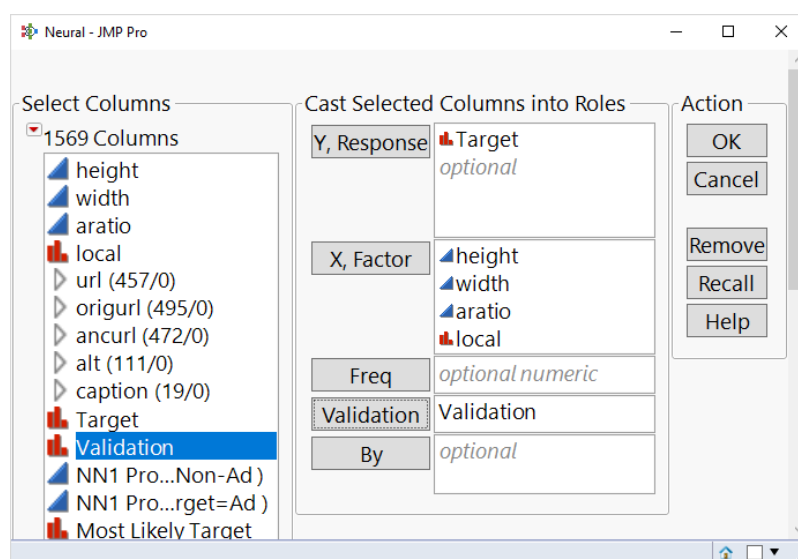
If data are a random sample, then a random subsample is OK. Sample from the level that is at the highest level of experimental units, so that observations within those units are independent. For example, sample so that all units within a block are in the same partition of the data. (Grouped Random option). If your data are time-sequenced, the partition should respect the time sequencing. (Cutpoint option)

Stratified Random is my first choice for most data. 2/3 training – 1/3 validation is customary, but not the rule. 1/4 validation is the minimum. More data -> larger validation.

In this case Stratified Random on Target makes sense to ensure balance across the response.

Neural Launch

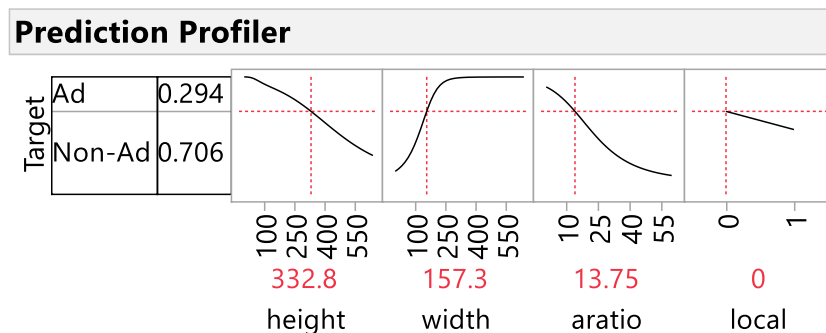
1. Select Analyze > Predictive Modeling > Neural.
2. Select Target column and click Y, Response.
3. Select Height through Local and click X, Factor.
4. Select your validation column and click Validation.



5. Click OK.
6. On the platform report window, click Go to perform automatic fitting using default settings.

Using the Neural model

- Look at Misclassification and Confusion Matrix for Training and for Validation to get a sense of how well the model is performing for the business task.
- Click the red triangle next to Model NTanH(3) and select Categorical Profiler.



- The profiler enables you to change the values of the predictors to obtain the predicted probability of Ad / No Add. You can use this to understand how the model classifies Ad and Non-Ad. Which predictors have the biggest effect?
- Click the red triangle next to Partition for Target and select Save Columns > Save Fast Formulas to save the model as formula columns in your data table.
- It is a good idea to rename your new formula columns for your reference. Double-click on the new column names in the Columns panel and add "NN1" at the start of the column name.

Target*	16,853,722	Non-Ad
Validation*	17,996,399	Non-Ad
NN1 Probability(Target=Non-Ad)		Non-Ad
NN1 Prob...arget=Ad)		Non-Ad

- Optional: Click the red triangle next to Partition for Target and select Save Columns > Publish Prediction Formula. This saves your model to a Formula Depot, which can be a convenient way to manage multiple models. We will use the Formula Depot later in this workshop.
- Save your data table (File menu > Save As). Do this after saving any model to the data table.

The process will be very similar when we look at other modelling methods in later modules.

Now you can go back to Model Launch and explore repeating this process with more and different nodes. Try Boosting by increasing Number of Models. Does the model improve (lower misclassification)?

You can also relaunch the platform and try adding more and different predictors.

Chris Gotwalt's Tips

Neural can be slow to fit when you have lots of predictors. You might want to use a small number of predictors in the first instance. Then when you have established the most important predictors using other methods in later sessions, you can use these predictors in a refined Neural model.

Hands-on session 2: Partition and Generalized Regression ML models

Partition Launch

1. Select Analyze > Predictive Modeling > Partition.
2. Select Target column and click Y, Response.
3. Select Height through Local and click X, Factor.
4. Select your validation column and click Validation.
5. Click OK.
6. On the platform report window, click Go to perform automatic splitting.

Using the Partition model

7. Click the red triangle next to Partition for Target and select Column Contributions. Which are the most important predictors?
8. Click the red triangle next to Partition for Target and select Profiler.
9. The profiler enables you to change the values of the predictors to obtain the predicted probability of Ad / No Add.
10. Click the red triangle next to Partition for Target and select Save Columns > Save Prediction Formula.
11. Rename formula columns for your reference as in session 1.
12. Save your data table.

Notice that the procedure is very similar to session 1.

Now you can explore relaunching the platform with more of the predictor variables and changing the Method (just go with default options in the first instance).

Save any models that you like.

Chris Gotwalt's Tips

It should be possible to fit a tree-based method fairly quickly, even with a large number of predictors. I sometimes use the column contributions to find the top predictors and then fit a Neural model with these.

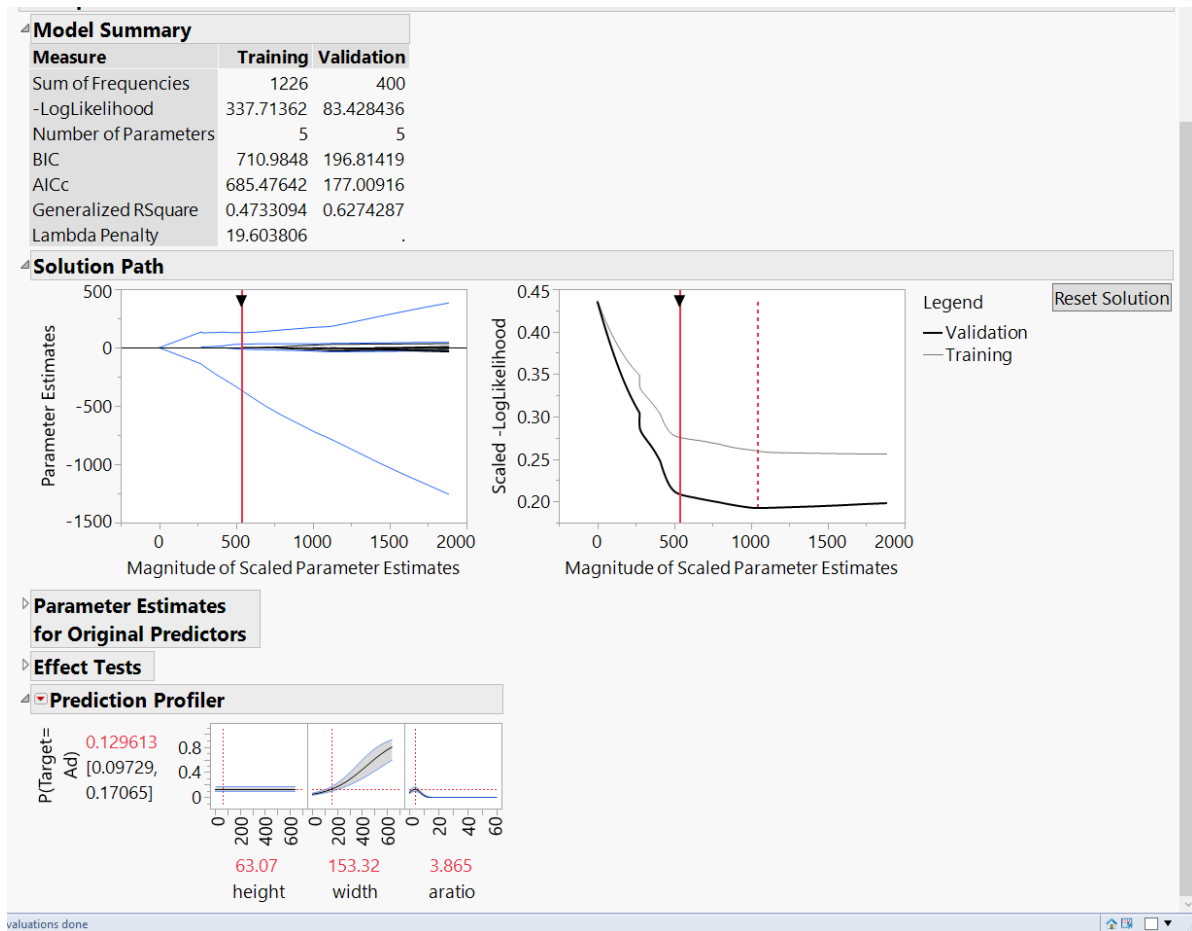
Generalized Regression Launch

1. Select Analyze > Fit Model.
2. Select Target column and click Y.
3. Select Height through Local and click Macros, then Response Surface.
4. Select your validation column and click Validation.
5. Change Personality to Generalized Regression and Target Level to 1.
6. Click Run.
7. On the platform report window, click Go to perform and Adaptive Lasso fitting.

Using the Generalized Regression model

8. Now look at the Solution Path. Hover the cursor over blue lines that extend most away from the 0 line. These are the most important predictor effects. Does this agree with what you found with other methods?
9. Minimise Parameter Estimates for Original Predictors by clicking the triangle next to the heading.
10. Click the red triangle next to Adaptive Lasso with Validation Column and select Profilers > Profiler.

- In the Solution Path drag the indicator (solid red vertical line with arrow) to the left or right to explore less and more complex models. See how the Profiler and Model Summary update. Can you find a simpler model that has a similar -LogLikelihood?



- Click the red triangle next to Adaptive Lasso with Validation Column and select Save Columns > Save Prediction Formula.
- Rename formula columns for your reference as in session 1.
- Save your data table.

Again, you can now explore alternatives in Model Launch and other predictors in the launch dialog.

Save any models that you like.

Chris Gotwalt's Tips

GenReg saves time and maximizes productivity: a little GenReg training goes a long way since the similarities between least squares, stepwise, logistic, robust, and censored models are so natural that you don't have to relearn options and an interface.

In predictive modeling exercises, one can use GenReg along with other modeling tools. Linear models have a moderate amount of predictive strength but are the most interpretable of all possible models. If you can achieve similar quality of fit between a neural model (for example) and a GenReg model, the GenReg model is recommended. This decision will often be subjective.

Hands-on session 3: Finding the best models

Optional: Models of models

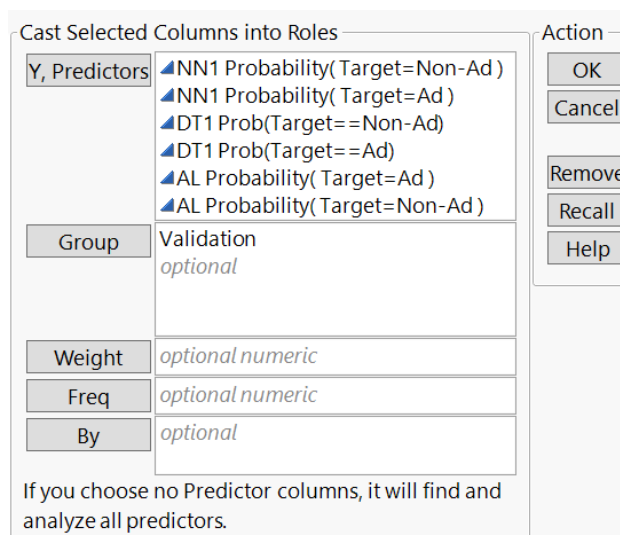
Try building a model-of-models using your save prediction formula columns as predictors.

Chris Gotwalt's Tips

Use the prediction formulas from earlier models as Xs, with the same validation column, and build a "model of models" predictor. Meta-models that handle multicollinearity work best. I recommend trying neural, PLS, and GenReg; avoid tree-based models. You can even combine the models back with the original input Xs.

Model Comparison

1. Select Analyze > Predictive Modeling > Model Comparison.
2. Now select the saved prediction columns that you wish to compare and click Y, Predictors.
3. Select your validation column and click Group.



4. Click OK.

Measures of Fit for Target										
Validation	Creator	.2468	Entropy	Generalized				Mean	Misclassification	
			RSquare	RSquare	Mean	-Log p	RMSE	Abs Dev	Rate	N
Training	Neural		0.4593	0.5673	0.2361	0.2657	0.1412		0.0865	1226
Training	Partition		0.4354	0.5353	0.2287	0.2556	0.1316		0.0813	1721
Training	Fit Generalized Adaptive Lasso		0.3644	0.4689	0.2801	0.2794	0.1637		0.0900	1233
Validation	Neural		0.5706	0.6735	0.187	0.2245	0.1144		0.0625	400
Validation	Partition		0.4675	0.5676	0.215	0.2349	0.1156		0.0662	574
Validation	Fit Generalized Adaptive Lasso		0.5131	0.6203	0.2131	0.2338	0.1394		0.0620	403

Which model has the lowest misclassification rate? You should only compare the performance on the Validation set.

How good is your 'best' model?

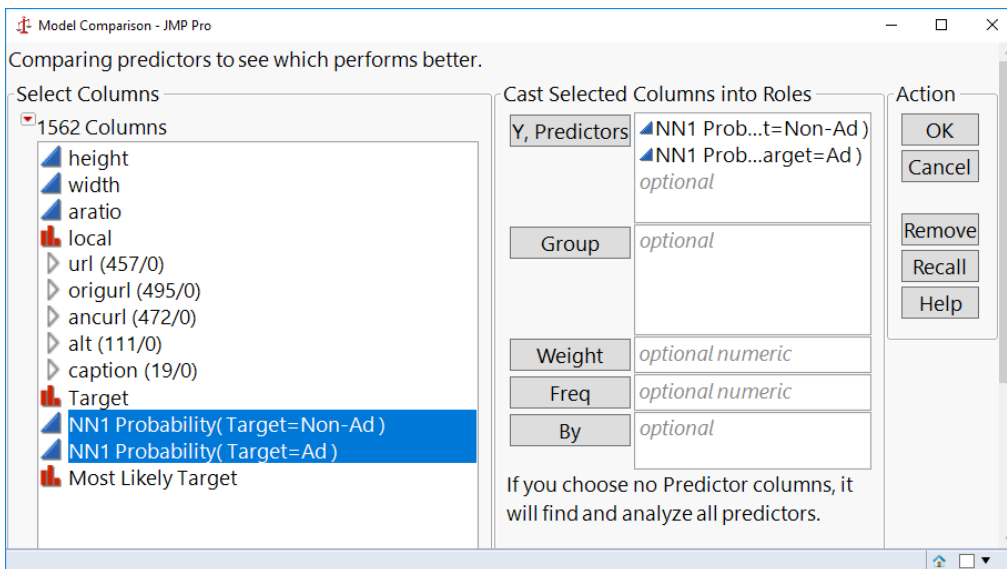
Now you will run your best model against the test data that has been held back. You will create a Formula Depot (if you didn't already in session 1) and add your best model. You can then apply the model to the test data once you have been given the password.

1. Launch the Formula Depot by selecting Analyze > Predictive Modeling > Formula Depot.

- Click the red triangle menu next to Formula Depot and select Add Formula from Column. A dialog box enables you to select the formula columns of interest. You only need select the “Most Likely...” saved prediction formula column for the model of interest.
- Optional: see how the options in the red triangle menu for the model in the Formula Depot enable you to translate the model for different environments.
- Now open the test data from the Add-In: Add-ins > The ‘AdFree’ study > Open ‘ad_free_data_training’.
- In the Formula Depot and the red triangle menu for the model select Run Script and then ‘ad_data_scoring’ to add the model formula columns to the test data.

You may find that by following these instructions the final model does not make predictions for some rows in the test data. This is because they have missing predictors. JMP Pro has capabilities to handle this but there may not be time to cover this in the workshop. Ask if you want to find out more.

- Launch Model Comparison from the ‘ad_data_scoring’ table, selecting the newly added Probability formula columns for your best model. (No need to specify a group variable in this case)



How did your model do in classifying Ad / Non-Ad in the test data? What is the misclassification rate and how does it compare with others in the workshop?

Model Comparison								
Predictors								
Measures of Fit for Target								
Creator	.2468	Entropy RSquare	Generalized RSquare	Mean -Log p	RMSE	Mean Abs Dev	Misclassification Rate	N
Neural		0.4876	0.5997	0.2329	0.2472	0.1332	0.0723	733