WEEK-1

1. Can H2O's GBM implementation work across multiple nodes (i.e. multiple computers, in a cluster)?

- (●) Yes, though very deep trees can cause communication overhead to become high.
- ( ) No, it only works on a single node, and you need to use random forest for clusters.
- ( ) Yes, though it becomes unstable if you set ntrees to over 50.

2. Bagging is ... (check all that can fit)

- [x] ...what random forest uses.
- [ ] ...what GBM uses.
- [ ] ...making new models that are influenced by the errors from previous models.
- [x] ...combining (e.g. averaging) the predictions of lots of models.

3. Boosting is ... (check all that can fit)

- [x] ...making new models that are influenced by the errors from previous models.
- [x] ...what GBM uses.
- [ ] ...what random forest uses.
- [ ] ...combining (e.g. averaging) the predictions of lots of models.

4. "*This change is likely to improve GBM model generalization, though you may need to increase the ntrees, to compensate.*"

Which of the following best fits that description?

(Hint: looking up parameters in the Appendix A of the H2O manual (

http://docs.h2o.ai/h2o/latest-stable/h2o-docs/index.html)

, or reading the GBM booklet (

http://docs.h2o.ai/h2o/latest-stable/h2o-docs/booklets/GBMBooklet.pdf [PDF])

, can help you answer this question.)

- ○ Decreasing max_depth from 5 to 3.
- ◉ Increasing max_depth from 5 to 10.
- ○ Experimenting with nbin_cats.
- ○ Decreasing learn_rate from 0.1 to 0.02.

1. What is the main disadvantage of using cross-validation instead of a validation data set

- ○ The model will have slightly lower accuracy on the training data.
- ○ The model no longer gives you a reliable estimate of accuracy.
- ◉ Training will take N times longer, where N is the number of folds.
- ○ The model will have slightly lower accuracy on unseen test data.

✓ **Correct**

2. If you want to do 8-fold cross-validation, which parameter do you add?

(To the estimator constructor in Python, to the learning algorithm function call in R.)

- ○ kfolds = 8
- ○ n =8
- ◉ nfolds = 8
- ○ cv = 8
- ○ cross_validation_folds = 8

3. If I use 5-fold cross-validation, giving a model id of "clever", then look on Flow, what model(s) will have been made?

- (●) Five models called "clever_cv_1" to "clever_cv_5", each trained on 80% of the training data, then one model called "clever", trained on 100% of the training data.
- ( ) Two models: "clever_cv" which stores the cross-validation results, and "clever" which is the model trained on 100% of the training data.
- ( ) A single model: "clever_cv", which is *an ensemble* of the five models, each trained on a different 80% of the training data.
- ( ) A single model: "clever", which *wraps* the five models, each trained on a different 80% of the training data; when using it the best performing of those 5 models will be used.

✓ **Correct**

4. When calling any of the performance functions on your model, what is the difference between using train, xval and valid as the argument?

- ( ) `train` evaluates on the training data, and you can use either `xval` or `valid` to get the performance on the unseen validation data (whether cross-validation or an explicit validation data set).
- (●) `train` gives the performance on the seen data, `xval` and `valid` give the performance on unseen data. Use `xval` if you specified cross-validation, use `valid` if you specified a validation set.
- ( ) E.g. if 5-fold cross-validation, `train` gives the average performance on the 80% of the data that was used for training, `xval` gives the average performance on the 20% that was kept aside, and `valid` gives the average performance on all 100% of the data.

5. What happens if you give a validation data set, and also specify 8-fold cross-validation?

(This was not covered in the lectures; either go and try it, or look in the documentation.)

- ○ You get an error message. It makes no sense to do this.
- ○ Each of the eight cv models is evaluated on both its fold of the training data, then again on the validation data, giving a more accurate estimate. The final model is built and evaluated in the same way, not using the validation data. Performance metrics for xval and valid will be identical.
- ○ Each of the eight cv models is evaluated on both its fold of the training data, then again on the validation data, giving a more accurate estimate. The final model is built and evaluated in the same way, not using the validation data. Performance metrics for train, xval and valid (probably) give different numbers.
- ◉ The eight cv models get built as normal, using their fold, but not using the validation data. Then the final model is built using 100% of the training data, but also uses the validation data. Performance metrics for train, xval and valid (probably) give different numbers.

✓ **Correct**

6. What happens if you request 1-fold cross validation?

(This was not covered in the video; either go and try it, or look in the documentation.)

- ○ You get an error message. It makes no sense to do this.
- ○ It runs, but does not do any cross-validation.
- ○ It runs, but does the default (10-fold cross validation).

6. What happens if you request 1-fold cross validation?

(This was not covered in the video; either go and try it, or look in the documentation.)

- ○ You get an error message. It makes no sense to do this.
- ○ It runs, but does not do any cross-validation.
- ○ It runs, but does the default (10-fold cross validation).
- ◉ It does the same as N-fold cross-validation, where N is the number of training rows in your data. I.e. each model builds on almost all the data, then is evaluated on a single row.

Incorrect
You can do this, but setting nfolds=1 is not the way. Instead you need to do nfolds=N. It will take a lot of time and memory, and gives diminishing returns in accuracy beyond about 10-fold.

7. In practical terms, how do you recognize over-fitting?

- ◉ When the score on your validation data stops improving or starts getting worse.
- ○ When the score on your training data stops improving or starts getting worse.
- ○ When the score on the validation data crosses over and becomes worse than on the training data.
- ○ When the score on the validation data crosses over and becomes better than on the training data.
- ○ When the gap between the score on training data and validation data grows larger than 10%.
- ○ When the score on the training data indicates it has started to learn the noise in the data.

Correct

8. Why did I increase max_depth from 5 to 10 when trying to over-fit a GBM model?

- (●) Deeper trees allow more complex rules to be fitted; I wanted it to be able to fit random noise.
- ( ) The amount of noise a GBM fits is proportional to the depth of the trees. That is why they are always shallow.
- ( ) Our `x` contains 4 columns, so 8 is the theoretical maximum useful depth, and it starts to overfit after that.
- ( ) 10 is the theoretical maximum for any GBM tree; any deeper is never useful.

✓ **Correct**

9. Why did I increase ntrees from 50 to 1000 when trying to over-fit a GBM model?

- (●) Each new tree in a GBM tries to fit the remaining error, which after a while is just the noise in the data set. I wanted to give it plenty of rope to hang itself with.
- ( ) You should increase ntrees by the square of the increase in max_depth. So 200 trees was enough, but I wanted to be sure.
- ( ) There was no need, just increasing max_depth was enough.

✓ **Correct**

WEEK-2

1. Can H2O's GBM implementation work across multiple nodes (i.e. multiple computers, in a cluster)?

- (●) Yes, though very deep trees can cause communication overhead to become high.
- ( ) No, it only works on a single node, and you need to use random forest for clusters.
- ( ) Yes, though it becomes unstable if you set ntrees to over 50.

2. Bagging is ... (check all that can fit)

- [x] ...what random forest uses.
- [ ] ...what GBM uses.
- [ ] ...making new models that are influenced by the errors from previous models.
- [x] ...combining (e.g. averaging) the predictions of lots of models.

3. Boosting is ... (check all that can fit)

- [x] ...making new models that are influenced by the errors from previous models.
- [x] ...what GBM uses.
- [ ] ...what random forest uses.
- [ ] ...combining (e.g. averaging) the predictions of lots of models.

4. "*This change is likely to improve GBM model generalization, though you may need to increase the ntrees, to compensat*

Which of the following best fits that description?

(Hint: looking up parameters in the Appendix A of the H2O manual (

http://docs.h2o.ai/h2o/latest-stable/h2o-docs/index.html)

, or reading the GBM booklet (

http://docs.h2o.ai/h2o/latest-stable/h2o-docs/booklets/GBMBooklet.pdf [PDF])

, can help you answer this question.)

- ◯ Decreasing max_depth from 5 to 3.
- ◯ Increasing max_depth from 5 to 10.
- ◯ Experimenting with nbin_cats.
- ◉ Decreasing learn_rate from 0.1 to 0.02.

1. What is the main disadvantage of using cross-validation instead of a validation data set?

   - ◯ The model will have slightly lower accuracy on the training data.
   - ◯ The model no longer gives you a reliable estimate of accuracy.
   - ◉ Training will take N times longer, where N is the number of folds.
   - ◯ The model will have slightly lower accuracy on unseen test data.

   ✓ **Correct**

2. If you want to do 8-fold cross-validation, which parameter do you add?

   (To the estimator constructor in Python, to the learning algorithm function call in R.)

   - ◯ kfolds = 8
   - ◯ n =8
   - ◉ nfolds = 8
   - ◯ cv = 8
   - ◯ cross_validation_folds = 8
   - ◯ k = 8

   ✓ **Correct**

3. If I use 5-fold cross-validation, giving a model id of "clever", then look on Flow, what model(s) will have been made? **1 / 1 point**

- (●) Five models called "clever_cv_1" to "clever_cv_5", each trained on 80% of the training data, then one model called "clever", trained on 100% of the training data.
- ( ) Two models: "clever_cv" which stores the cross-validation results, and "clever" which is the model trained on 100% of the training data.
- ( ) A single model: "clever_cv", which is *an ensemble* of the five models, each trained on a different 80% of the training data.
- ( ) A single model: "clever", which *wraps* the five models, each trained on a different 80% of the training data; when using it the best performing of those 5 models will be used.

✓ **Correct**

4. When calling any of the performance functions on your model, what is the difference between using train, xval and valid as the argument? **1 / 1 point**

- ( ) `train` evaluates on the training data, and you can use either `xval` or `valid` to get the performance on the unseen validation data (whether cross-validation or an explicit validation data set).
- (●) `train` gives the performance on the seen data, `xval` and `valid` give the performance on unseen data. Use `xval` if you specified cross-validation, use `valid` if you specified a validation set.
- ( ) E.g. if 5-fold cross-validation, `train` gives the average performance on the 80% of the data that was used for training, `xval` gives the average performance on the 20% that was kept aside, and `valid` gives the average performance on all 100% of the data.

✓ **Correct**

5. What happens if you give a validation data set, and also specify 8-fold cross-validation?

(This was not covered in the lectures; either go and try it, or look in the documentation.)

- ○ You get an error message. It makes no sense to do this.
- ○ Each of the eight cv models is evaluated on both its fold of the training data, then again on the validation data, giving a more accurate estimate. The final model is built and evaluated in the same way, not using the validation data. Performance metrics for xval and valid will be identical.
- ○ Each of the eight cv models is evaluated on both its fold of the training data, then again on the validation data, giving a more accurate estimate. The final model is built and evaluated in the same way, not using the validation data. Performance metrics for train, xval and valid (probably) give different numbers.
- ◉ The eight cv models get built as normal, using their fold, but not using the validation data. Then the final model is built using 100% of the training data, but also uses the validation data. Performance metrics for train, xval and valid (probably) give different numbers.

✓ **Correct**

7. In practical terms, how do you recognize over-fitting?

- (●) When the score on your validation data stops improving or starts getting worse.
- ( ) When the score on your training data stops improving or starts getting worse.
- ( ) When the score on the validation data crosses over and becomes worse than on the training data.
- ( ) When the score on the validation data crosses over and becomes better than on the training data.
- ( ) When the gap between the score on training data and validation data grows larger than 10%.
- ( ) When the score on the training data indicates it has started to learn the noise in the data.

✓ **Correct**

8. Why did I increase max_depth from 5 to 10 when trying to over-fit a GBM model?

- (●) Deeper trees allow more complex rules to be fitted; I wanted it to be able to fit random noise.
- ( ) The amount of noise a GBM fits is proportional to the depth of the trees. That is why they are always shallow.
- ( ) Our `x` contains 4 columns, so 8 is the theoretical maximum useful depth, and it starts to overfit after that.
- ( ) 10 is the theoretical maximum for any GBM tree; any deeper is never useful.

✓ **Correct**

9. Why did I increase ntrees from 50 to 1000 when trying to over-fit a GBM model?

- (●) Each new tree in a GBM tries to fit the remaining error, which after a while is just the noise in the data set. I wanted to give it plenty of rope to hang itself with.
- ( ) You should increase ntrees by the square of the increase in max_depth. So 200 trees was enough, but I wanted to be sure.
- ( ) There was no need, just increasing max_depth was enough.

✓ **Correct**

WEEK-3

1. The difference between uploading and importing a file is:

- ○ Imports are kept in the memory of your client, uploads are kept in memory of the h2o server machine, so imports are quicker.
- ○ Uploads are kept on disk on the h2o server machine, while imports are kept in memory.
- ○ No difference - they both call the same code.
- ◉ Imports are loaded from local or remote file systems the h2o server can see; uploads are loaded from the local file system of your client (by first doing a file transfer to the h2o server, then an import).

Correct
Yes. Prefer import to upload, assuming the H2O server can see the file.

2. h2o.import_file() (h2o.importFile() in R), have optional parameters to do which of the following?

(check all that apply)

(Go and look at the required and optional arguments, if you haven't already!)

- ☑ Allow you to specify the name of the frame that is created.

Correct
Yes. Or you can use h2o.assign() afterwards.

- ☐ Control the distribution of the data over your H2O cluster.
- ☑ Specify the column names.

Correct
This is useful if the first row is data, and not column names.

- ☑ Specify the column types.

Correct
The alternative is to check and fix column types after loading.

- ☑ Specify which strings should be treated as NAs (missing data).

1. Select all of the following can be used as the "family" parameter for predicting species in the iris data set, when using GLM.

- [ ] gaussian
- [ ] quasibinomial
- [ ] tweedie
- [x] multinomial

Correct
Predicting species in the iris data set is a classification problem, with three choices.

- [ ] poisson
- [ ] gamma
- [ ] binomial

2. Linear models can be solved analytically (meaning no iteration required, and the solver parameter is ignored), under certain conditions.

From the below select where GLM will solve it analytically. Assume it is a regression. Check all that apply.

- [ ] family = "binomial", alpha = 0
- [ ] family = "binomial", alpha = 0.5, lambda = 0
- [ ] family = "gaussian", alpha = 0.2, lambda = 1e-5
- [x] family = "gaussian", alpha = 1.0, lambda = 0

Correct
Zero lambda means no regularization, which allows solving analytically.

- [x] family = "gaussian", alpha = 0.0, lambda = 1e-5

Correct
alpha=0 means ridge regression (i.e. only L2 regularization), which allows solving analytically.

3. If lambda is set explicitly, but set too high, what happens?

- (•) All coefficients go to zero and model produced is not much use.
- ( ) An exception is thrown, but it is hard to know in advance what value counts as "too high".
- ( ) All kinds of weirdness.

Correct
If all coefficients are zero, it means it gives zero weight to all the input columns.

1. When loading csv files, which of these are true. Check all that apply.

- [ ] The top row is always column names. Insert a blank line if you want default names to be chosen.
- [x] The top row can be column names, but this is optional. It will analyze the data to decide if the 1st row is data or not.
- [ ] You can specify column types in the second row, optionally. It will analyze the data to decide if the 2nd row is data or not.
- [x] Columns are analyzed to decided the type. You should check, and manually fix the column type, if it gets it wrong.
- [x] zip and gzip files are handled automatically.

2. In GLM, an "alpha" of zero (with the default value for "lambda") means:

- (●) Ridge regression, where lambda is the L2 penalty, meaning all coefficients are shrunk towards zero.
- ( ) No regularization is performed.
- ( ) A balance of ridge and lasso regression, also called Elastic Net.
- ( ) Lasso regression, where lambda is the L1 penalty, meaning some coefficients become zero and are effectively removed from the model.

3. If our "y", our response variable, is an integer representing the number of times something happens, which setting for "family" is best, when using GLM?

- (●) poisson
- ( ) multinomial
- ( ) binomial

4. If you wish to use grid search in H2O, how do you do it?

- ( ) You set up a series of nested loops, one for each hyper-parameter you wish to use, with your code to make an H2O model in the innermost loop.
- (●) In R, use h2o.grid(). In Python create an object of h2o.grid.H2OGridSearch(), and then call train() on that object.
- ( ) In R, add the argument "grid=TRUE" to your model, and in Python it is almost the same: "grid=True".

5. What is the advantage of using H2O's "cor()" compared to bringing your data into your R or Python session and using the correlation function there?

- ○ It allows you run on data too big to fit in the memory of a single machine; it also saves you that download.
- ◉ It has a higher precision
- ○ It automatically works with NAs.

6. If you have "mydata.csv.gz", which is gzipped csv data, what do you have to do load it into H2O?

- ○ Convert it to a zip format, and use h2o.importFile("mydata.csv.zip")
- ○ Use gzip -d, to uncompress it, then h2o.importFile("mydata.csv")
- ◉ Use h2o.importFile("mydata.csv.gz"). It will deal with it correctly.

7. If you wanted to save an H2O data frame to Amazon S3, which of these would you give to export_file() (exportFile() in R)

- ○ Add an entry in core-site.xml, and use whatever label you used there.
- ◉ "s3n://mybucket/mydata.csv"
- ○ "aws+mybucket/mydata.csv"

8. Which type of models can naive bayes NOT build?

(There are two correct answers, select both.)

- ☑ Regression
- ☑ Unsupervised
- ☐ Binomial classification
- ☐ Multinomial classification

WEEK-4

1. Early stopping mainly helps guard against what?

- (●) Over-fitting
- ( ) Excessive power usage
- ( ) Numeric instability

Correct
Yes, it stops the model training once it looks like it is not learning any more.

2. Why is it better to use a validation data set, or cross-validation, when using early stopping?

- ○ H2O might crash otherwise.
- ○ When there is no validation data, but early stopping is being used, H2O will syphon off some of your data for validation, but this means your model will perform more poorly.
- ◉ If no validation data, it will use the score on the training data to judge early stopping. But it is likely to keep improving on that data, so early stopping might never trigger.

✓ **Correct**
You really need some data it is not training on, in order to be able to judge over-fitting. However, in some situations (such as setting threshold quite high) early stopping without validation data might make sense.

3. Mr.Cook complains: "I compared with and without early stopping, and they both ran for the same amount of time, and the model without early stopping got a better score!".

Which explanation is best?

- ○ That is impossible. He must have a bug somewhere else in his code.
- ○ Sometimes there is a bug in early stopping. Maybe he should try restarting H2O.
- ◉ The model was still improving at the time it stopped. You need to run it for longer to see early stopping have an effect. Random variation from model to model was behind the better score.
- ○ Mr.Cook is always finding something to moan about. I hate working with him.

✓ **Correct**
Yes, this is the most likely explanation.

1. What restriction is there when using h2o.cbind(A, B) ? (A.cbind(B) in Python)

- ○ B must have at least one column in common with A.
- ○ Both A and B must have the same number of columns.
- ○ B must have a subset of A's columns
- ◉ Both A and B must have the same number of rows.

Correct
Yes, to bind columns they must have the same number of rows.

2. What restriction is there when using h2o.rbind(A, B) ?

- ◉ Both A and B must have the same number of columns.
- ○ Both A and B must have the same number of rows.
- ○ B must have a subset of A's columns
- ○ B must have at least one column in common with A.

Correct
Not just the same number, but the same names and column types, too.

1. What mini-batch size is used, by default, in training?

   (Mini batch refers to how many training samples are processed at a time.)

   - ○ It depends on the number of training samples.
   - ○ It depends on how many epochs you specify.
   - ○ 10
   - ◉ 1

   Correct
   Yes, the weights are updated after every single training sample. Look up the HOGWILD! algorithm for how this is possible. However there is a mini_batch_size parameter you can set if you want to experiment with other values.

2. Increasing epochs has what effect(s)? Check all that apply.

   - ☐ The average weight of the neurons in the final hidden layer will be lower.
   - ☑ It will take longer to run. (Unless early stopping stops it.)

   Correct
   Each epoch is one pass through all the training data.

   Once early stopping criteria are reached, training will stop, even if not all the epochs have been done.

   - ☐ The average weight of the neurons in the first hidden layer will be lower.
   - ☑ The model might perform better, but it might also over-fit more.

   Correct
   Yes, more epochs means more time spent training, which means the weights get more chance to fit the training data.

3. How many epochs are done by default?

- ○ The square root of the number of training samples (rounded up)
- ◉ 10
- ○ 1
- ○ There is no default, it is a required parameter.
- ○ The square root of the number of training samples (rounded down)

✓ **Correct**

1. The default setting for activation is what?

- (●) Rectifier
- ( ) Tanh
- ( ) Maxout
- ( ) RectifierWithDropout
- ( ) MaxoutWithDropout
- ( ) TanhWithDropout

Correct
Yes. You need to change this if you want to specify hidden dropout ratios (but not if you just want to specify an *input* dropout ratios).

2. How do you specify L1 and L2 regularization to both be 0.00001 ?

- (●) l1=1e-5, l2=1e-5
- ( ) You can't: L1 or L2, but not both.
- ( ) alpha = 0.5, lambda = 1e-5
- ( ) l1l2 = 1e-5

Correct
The 1e-5 is a useful shorthand, that can be used in both R and Python. It is harder to make a mistake with.

WEEK-5

1. Autoencoders are related to which other h2o algorithm?

- ○ All of the above.
- ◉ Deep learning
- ○ GBM
- ○ GLM

Correct
And almost all the same parameters can be used.

2. The MSE of an autoencoder is representing what?

- ◉ The amount of information being lost between input and output.
- ○ The amount of data compression
- ○ The square of the amount of data compression.

Correct
You might also think of it as the amount of noise being introduced by the network.

Note: there are not really any real-world units involved, so no advantage to using RMSE or MAE over MSE.

3. A stacked autoencoder refers to what?

- ◉ Two or more autoencoders, the output of the first becoming the input of the second. (Where "output" is referring to the values from the middle hidden layer.)
- ○ An autoencoder using 2 or more hidden layers
- ○ An autoencoder using 3, 5 or any odd number of hidden layers, where they are symmetrical. (e.g. 50,10,50, but not 50,10,40)
- ○ An ensemble of auto-encoders.
- ○ A stacked ensemble of auto-encoders.

Correct
The autoencoders are stacked one after the other, hence the term.

4. Which is usually the best activation function to use with autoencoders, and why?

- ○ Rectifier (or RectifierWithDropout if using dropout), as it is quicker.
- ○ Maxout, as it is designed for autoencoders; also you should never use dropout with autoencoders.
- ○ RectifierWithDropout, as it is quicker, and you should alway use at least a small amount of dropout with autoencoders.
- ◉ Tanh (or TanhWithDropout if using dropout), as it is more numerically stable than the alternatives.

Correct
Rectifier can often lead to some very high weights, and some weights that are zero.

4. Which is usually the best activation function to use with autoencoders, and why?

- ○ Rectifier (or RectifierWithDropout if using dropout), as it is quicker.
- ○ Maxout, as it is designed for autoencoders; also you should never use dropout with autoencoders.
- ○ RectifierWithDropout, as it is quicker, and you should alway use at least a small amount of dropout with autoencoders.
- ◉ Tanh (or TanhWithDropout if using dropout), as it is more numerically stable than the alternatives.

✓ **Correct**
Rectifier can often lead to some very high weights, and some weights that are zero.

5. What is the maximum number of hidden layers you can use in an autoencoder?

- ○ 1
- ○ 2
- ○ 3
- ○ Any odd number
- ◉ There is no limit

✓ **Correct**
Yes. Using an odd number and being symmetrical is recommended, but not required.

1. You want to divide your customer database, of about 500 records, into three approximately equal-sized clusters, based on all their interests. What problems are there with using kmeans for this? Check all that apply.

- [x] kmeans can easily create 3 clusters, but there is no way to specify their size.
- [ ] kmeans needs big data, much more than 500 records, to work effectively.
- [ ] kmeans will be very slow, even on just 500 records.

2. You wish to visualize your data as a 2D scatter plot. Your data has 300 numeric columns. Which of the following are a reasonable approach?

- [x] Build an autoencoder that has a single layer of just two hidden nodes. Use deepfeatures() to extract the value in those two hidden nodes for each row in your data, and plot those values.
- [ ] Use H2O's anomaly function, and plot based on the reconstruction error.
- [x] Use principal components analysis, and take the first two components (using predict) and plot them.
- [ ] Use kmeans, with k set to 2, and plot them.

3. You have a record in your data (i.e. a sample in your training data) that you are interested in and you want to know what records in your data are similar to it. Which of the following are reasonable approaches?

- [x] Use h2o.kmeans(), see which cluster your record of interest is in, and look at the others in the same cluster.
- [x] Use any dimension reduction technique, with just one dimension. Get the value for your record of interest and then look at the records which have the closest value (above or below) to it.
- [ ] Use h2o.anomaly(), and sort the records by their reconstruction error. Look at those records which are adjacent to your record of interest in the sorted list.

4. Which is the more accurate statement?

- ( ) K-means offers a good solution for the curse of dimensionality.
- (•) The curse of dimensionality is a problem for k-means.

5. Say you reduce a dataset, using an autoencoder to N dimensions. How do those dimensions differ from the first N components returned by h2o.prcomp() ?

- ( ) They don't really differ: given enough time and resources the two algorithms will return identical results.
- ( ) The 1st autoencoder dimension (the first neuron in the hidden layer) carries more information than the 2nd, which carries more than the 3rd and so on. The components in PCA can all be considered to carry equal weight.
- (•) The autoencoder-produced N dimensions all carry equal amounts of information, whereas in PCA each dimension (component) carries more information than the dimension that follows.

1. Which of these algorithms in H2O is based on deep learning. Check all that apply.

- [x] autoencoder
- [ ] GLRM
- [ ] prcomp
- [ ] kmeans
- [x] anomaly

2. Which of these is worth considering to patch the 50% missing data values in a column, which you know has a high correlation with a number of other columns?

(Hint: more than one)

- [ ] prcomp
- [x] deep learning
- [x] GLM
- [x] random forest
- [ ] anomaly

3. The oldest records are missing a value in 3 columns; this represents about 1% of the data. Those 3 columns correlate well with each other but very little with the other columns, including our response variable. Which of these approaches are reasonable?

Choose all that apply.

- [x] Drop those rows from the data frame.
- [x] Drop those three columns from the data frame.
- [x] Do nothing, and let each algorithm deal with it.

4. An autoencoder with a single layer of 200 hidden neurons, which is then fed into a supervised deep learning model (using all defaults, i.e. two hidden layers with 200 neurons in each), is found to have superior accuracy compared to just feeding the original data directly into that same default deep learning model. Which is the best explanation?

- (•) It is very similar to having three hidden layers, each of 200 neurons.
- ( ) The autoencoder will smooth out the data and make it quicker for the supervised deep learning model to run.
- ( ) It makes no sense, and is almost certainly random variation (luck) or a bug in the experimenter's code.

5. In some of this week's coding, we explicitly set train_samples_per_iteration, score_interval and score_duty_cycle when experimenting with autoencoders. Why?

- (•) Because it was a small dataset, this allowed us to see the way MSE changes after each epoch.
- ( ) This is an advanced technique to make deep learning (not just autoencoders) more efficient, and you should generally do this.
- ( ) It is something you have to do when setting activation to Tanh.
- ( ) This is recommended behaviour for autoencoders (but not supervised deep learning).

6. In the Hands-On Data Repair video, we used gamma distribution to repair CarrierDelay. The default, gaussian, gave some negative values. Which is the best explanation why?

- (●) Gaussian assumes values are evenly distributed either side of the mean. Because there were so many zeroes in the good training data, the mean is only just above zero.
- ( ) The data is very long-tailed.
- ( ) The good training data had values that were both above and below zero, and the gaussian model was emulating that, but gamma didn't.

7. Which H2O algorithm(s) offer a missing_data_handling argument?

- [x] GBM
- [x] Deep Learning
- [ ] Random Forest
- [ ] GLM

8. If you are using an autoencoder, with 8 inputs, which of these architectures is likely to give the *lowest* MSE?

- ( ) One hidden layer, with just a single neuron.
- ( ) Three hidden layers, organized as 4-8-4
- ( ) Three hidden layers, organized as 8-4-8
- (●) One hidden layer of eight neurons.

9. Which of these is an example of a stacked autoencoder?

- [x] An autoencoder with a single hidden layer of 8 neurons. Using deepfeatures on the training data gives another data set, which is fed into another autoencoder with a single hidden layer of 4 neurons.
- [ ] An autoencoder with 2+ hidden layers where each layer has the same number of neurons.
- [ ] An autoencoder with three hidden layers, with 2-4-8 neurons, respectively.
- [ ] An autoencoder with three hidden layers, with 8-4-2 neurons, respectively.

10. When building an autoencoder with more hidden neurons than the number of input dimensions, it is common to use dropouts. With that in mind, which is the best activation function to use?

- [ ] The default: Rectifier
- [ ] RectifierWithDropout
- [ ] Tanh
- [x] TanhWithDropout

WEEK-6

1. Which is the best definition of an ensemble model, from the following?

- ○ A neural network model, that uses at least three hidden layers.
- ○ A model for doing better in pub quizzes, and other NLP tasks.
- ◉ Using a number of models together, for the purpose of getting a better prediction.
- ○ A set of models that work in harmony together.

Correct
Yes.

2. Which is the best description of the stacked ensemble algorithm?

- ◉ It takes a set of models, and uses another model to learn the best way to combine their predictions.
- ○ It uses bagging along with stacked autoencoders.
- ○ It is a black box, a bit like deep learning, but even blacker. It might even involve magic.

Correct
Yes: it is that additional model to combine predictions that make it a stacked ensemble.

3. To use stacked ensembles, as implemented in H2O, what is the requirement on each individual model? Choose all that apply.

- ☑ Each cv fold must be identical for each model (e.g. by using modulo, or a random seed).

Correct
This is important for the algorithm to work correctly.

- ☑ Each must have used exactly the same training data.

Correct
This is important for the algorithm to work correctly.

- ☑ It is supervised learning.

Correct
Stacked ensembles are a supervised learning technique.

- ☑ It must have been built with cross-validation, and the predictions kept.

Correct
This data is needed by the algorithm.

- ☐ Each must have used a common prefix on the model id.

4. Given that you've made some models (trained using "x", "y" and "train", with nfolds=5), and put their IDs into "model_ids", how do you make a stacked ensemble in H2O?

- ○ In R: m <- h2o.stackedEnsemble(x, y, train, nfolds = 5, base_models = model_ids)

  In Python:

  ```
  from h2o.estimators.stackedensemble import H2OStackedEnsembleEstimator
  m = H2OStackedEnsembleEstimator(base_models=models, nfolds=5)
  m.train(x, y, train)
  ```

- ◉ In R: m <- h2o.stackedEnsemble(x, y, train, base_models = model_ids)

  In Python:

  ```
  from h2o.estimators.stackedensemble import H2OStackedEnsembleEstimator
  m = H2OStackedEnsembleEstimator(base_models=models)
  m.train(x, y, train)
  ```

- ○ In R: m <- h2o.stackedEnsemble(base_models = model_ids)

  In Python:

  ```
  from h2o.estimators.stackedensemble import H2OStackedEnsembleEstimator
  m = H2OStackedEnsembleEstimator(base_models=models)
  m.train()
  ```

✓ **Correct**
Yes. Give the same x,y,train, and the list of models to use.