# Machine Learning with $H_2O$

$H_2O$.ai

# H2O Algorithms
# Un-supervised

H2O.ai

# H2O Algorithms

## *Supervised Learning*

**Statistical Analysis**

- **Generalized Linear Models**: Binomial, Gaussian, Gamma, Poisson, and Tweedie
- **Naive Bayes**: Binary Text Classification

**Ensembles**

- **Distributed Random Forest**: Classification or Regression Models
- **Gradient Boosting Machine**: Ensembles of shallow decision trees with increasing refined approximations

**Deep Neural Networks**

- **Deep Learning**: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

# H2O Algorithms

## *Unsupervised Learning*

**Clustering**

- **K-means**: Partition observations into k clusters of the same spatial size. Categorical features are one hot encoded.
- **Archetypes [GLRM]**: Partition observations into k archetypes.

**Dimensionality Reduction**

- **Principal Component Analysis**: Linearly transforms correlated variables to independent components
- **Generalized Low Rank Model:** Approximates data set as a product of two low dimensional factors. Extends PCA to handle sparse data, categorical data, and adds regularization.

**Anomaly Detection**

- **Autoencoders [Deep Learning]**: Create multi-layer feed forward neural networks starting with an input layer followed by multiple layers of nonlinear transformations

# Unsupervised Learning Models

- No target!
- Ways to characterize or simplify data

- Uses
  - Clustering
  - Variable reduction
  - Compression
  - Anomaly detection

Unsupervised Learning:

# K-MEANS CLUSTERING

# Clustering

- We are going to try and “bundle” customers/companies/transactions based on “likeness” of certain features.
- These “clusters” will help us simplify and potentially “segment” observations.
  - Segment customers
  - Categorize events

# K-Means in H2O

CS `buildModel "kmeans"`

12ms

## Build a Model

Select an algorithm: K-means

*PARAMETERS* — *GRID?*

| | | |
|---|---|---|
| *model_id* | kmeans-bdc9ae94-5337-460d-9692-7460bdbb429a | Destination id for this model; auto-generated if not specified. |
| *training_frame* | (Choose...) | Id of the training data frame (Not required, to allow initial validation of model parameters). |
| *validation_frame* | (Choose...) | Id of the validation data frame. |
| *nfolds* | 0 | Number of folds for N-fold cross-validation (0 to disable or >= 2). |
| *ignored_columns* | Search... | |

All | None | Previous 100 | Next 100

Only show columns with more than 0 % missing values.

| | | |
|---|---|---|
| *ignore_const_cols* | ☑ | Ignore constant columns. |
| *k* | 1 | The max. number of clusters. If estimate_k is disabled, the model will find k centroids, otherwise it will find up to k centroids. |
| *estimate_k* | ☐ | Whether to estimate the number of clusters (<=k) iteratively and deterministically. |
| *max_iterations* | 10 | Maximum training iterations (if estimate_k is enabled, then this is for each inner Lloyds iteration) |
| *standardize* | ☑ | Standardize columns before computing distances |
| *init* | Furthest | Initialization mode |

*ADVANCED* — *GRID?*

| | | |
|---|---|---|
| *fold_assignment* | AUTO | Cross-validation fold assignment scheme, if fold_column is not specified. The 'Stratified' option will stratify the folds based on the response variable, for classification problems. |
| *fold_column* | (Choose...) | Column with cross-validation fold index assignment per observation. |
| *score_each_iteration* | ☐ | Whether to score during each iteration of model training. |
| *seed* | -1 | RNG Seed |
| *max_runtime_secs* | 0 | Maximum allowed runtime in seconds for model training. Use 0 to disable. |
| *categorical_encoding* | AUTO | Encoding scheme for categorical features |

*EXPERT* — *GRID?*

| | | |
|---|---|---|
| *keep_cross_validation_predictions* | ☐ | Whether to keep the predictions of the cross-validation models. |
| *keep_cross_validation_fold_assignment* | ☐ | Whether to keep the cross-validation fold assignment. |
| *user_points* | (Choose...) | This option allows you to specify a dataframe, where each row represents an initial cluster center. The user-specified points must have the same number of columns as the training observations. The number of rows must equal the number of clusters |

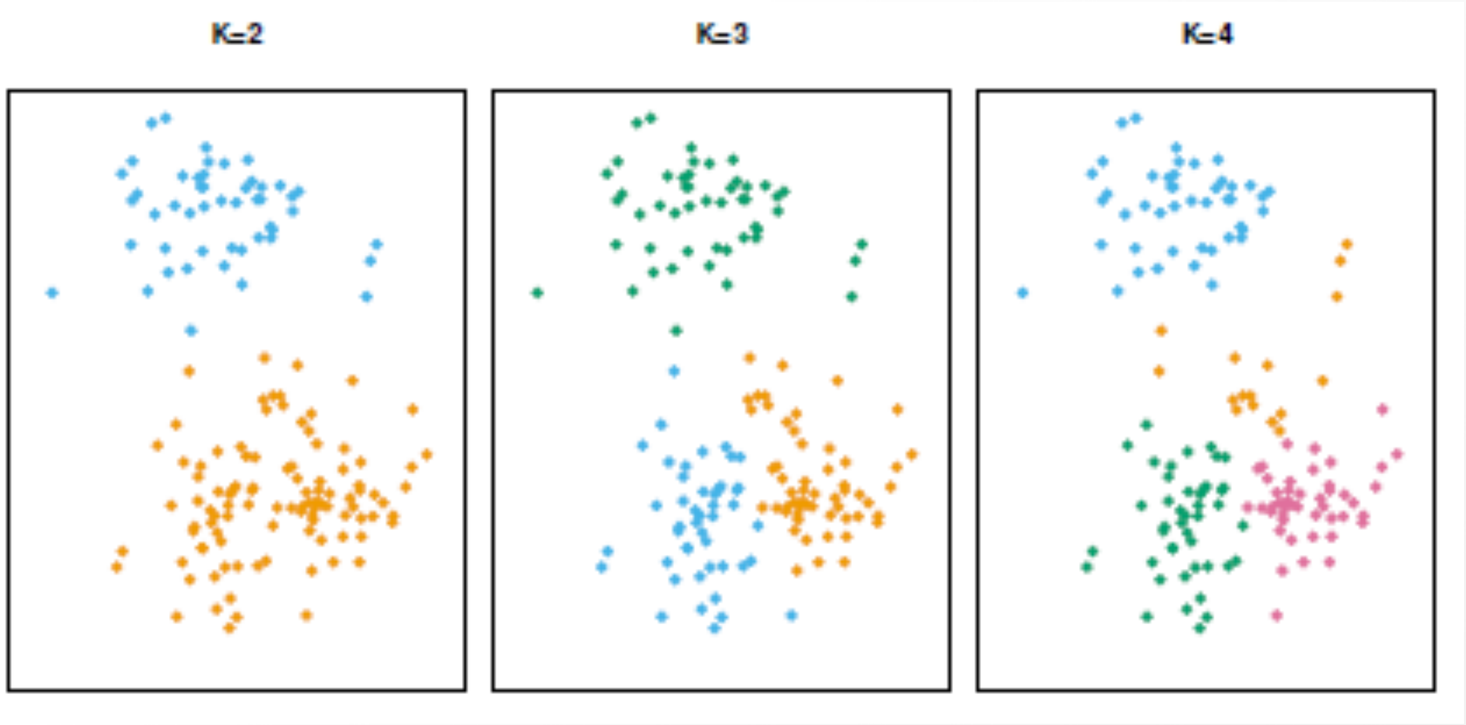
Build Model

# Clustering

Objective: Minimize the within cluster variation for a given k

How to choose k?

- Choose several and evaluate performance
- Often business rules determine feasible k



# What is Principal Components Analysis

- Another method of data simplification
- Can I reduce the width (dimensionality) of the dataset by creating one or more new PCA variables that do a “good job” of representing the original signal in the data?
- Uses:
  - Variable reduction=> Fewer predictors
  - Feature extraction=> Anomaly detection

# Principal Component Analysis in H2O

CS `buildModel "pca"`

9ms

## Build a Model

Select an algorithm: Principal Components Analysis

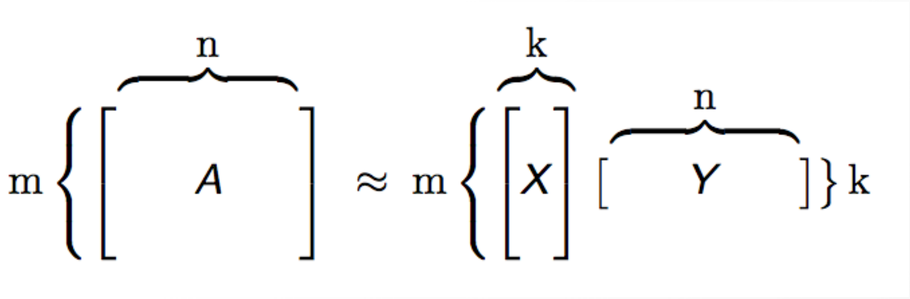
| PARAMETERS | | | GRID? |
|---|---|---|---|
| *model_id* | pca-5385d795-5fb4-466f-b884-c703a6a89424 | Destination id for this model; auto-generated if not specified. | |
| *training_frame* | (Choose...) | Id of the training data frame (Not required, to allow initial validation of model parameters). | |
| *validation_frame* | (Choose...) | Id of the validation data frame. | ☐ |
| *ignored_columns* | Search... | | |
| | All / None | Previous 100 / Next 100 | |
| | Only show columns with more than 0 % missing values. | | |
| *ignore_const_cols* | ☑ | Ignore constant columns. | |
| *transform* | NONE | Transformation of training data | ☐ |
| *pca_method* | GramSVD | Method for computing PCA (Caution: GLRM is currently experimental and unstable) | |
| *k** | 1 | Rank of matrix approximation | ☐ |
| *max_iterations* | 1000 | Maximum training iterations | ☐ |
| *use_all_factor_levels* | ☐ | Whether first factor level is included in each categorical expansion | |
| *compute_metrics* | ☑ | Whether to compute metrics on the training data | |
| *impute_missing* | ☐ | Whether to impute missing entries with the column mean | |
| *seed* | -1 | RNG seed for initialization | |

| ADVANCED | | | GRID? |
|---|---|---|---|
| *score_each_iteration* | ☐ | Whether to score during each iteration of model training. | |
| *max_runtime_secs* | 0 | Maximum allowed runtime in seconds for model training. Use 0 to disable. | ☐ |

Build Model

# Generalized Low Rank Models

- **Given:** Data table A w/ m rows and n columns
- **Find:** Compressed representation as numeric table X and Y where # cols in X = # rows in Y = small user specified K << max(m,n)
- # cols in Y is d = (total dimension of embedded features in A) ≥ n




- Row of Y = archetypal fea
- Row of X = row of A in reduced feature space
- Can approximately reconstruct A from product XY

# Generalized Low Rank Modeling in H2O

CS `buildModel "glrm"`

7ms

## Build a Model

Select an algorithm: Generalized Low Rank Modeling

| PARAMETERS | | | GRID? |
|---|---|---|---|
| *model_id* | glrm-767dc939-83ed-4ce7-af3f-37c084f9a76b | Destination id for this model; auto-generated if not specified. | |
| *training_frame* | (Choose...) | Id of the training data frame (Not required, to allow initial validation of model parameters). | |
| *validation_frame* | (Choose...) | Id of the validation data frame. | ☐ |
| *ignored_columns* | Search... | | |
| | ☑ All ☐ None | ← Previous 100 → Next 100 | |
| | Only show columns with more than 0 % missing values. | | |
| *ignore_const_cols* | ☑ | Ignore constant columns. | |
| *loading_name* | | Frame key to save resulting X | |
| *transform* | NONE | Transformation of training data | ☐ |
| *k** | 1 | Rank of matrix approximation | ☐ |
| *loss* | Quadratic | Numeric loss function | ☐ |
| *loss_by_col_idx* | | Loss function by column index (override) | |
| *multi_loss* | Categorical | Categorical loss function | ☐ |
| *period* | 1 | Length of period (only used with periodic loss function) | ☐ |
| *regularization_x* | None | Regularization function for X matrix | ☐ |
| *regularization_y* | None | Regularization function for Y matrix | ☐ |
| *gamma_x* | 0 | Regularization weight on X matrix | ☐ |
| *gamma_y* | 0 | Regularization weight on Y matrix | ☐ |
| *max_iterations* | 1000 | Maximum number of iterations | ☐ |
| *max_updates* | 2000 | Maximum number of updates, defaults to 2*max_iterations | ☐ |
| *init_step_size* | 1 | Initial step size | ☐ |
| *min_step_size* | 0.0001 | Minimum step size | ☐ |
| *seed* | -1 | RNG seed for initialization | ☐ |
| *init* | PlusPlus | Initialization mode | ☐ |
| *svd_method* | Randomized | Method for computing SVD during initialization (Caution: Randomized is currently experimental and unstable) | ☐ |
| *user_y* | (Choose...) | User-specified initial Y | |
| *user_x* | (Choose...) | User-specified initial X | |
| *expand_user_y* | ☑ | Expand categorical columns in user-specified initial Y | |
| *impute_original* | ☐ | Reconstruct original training data by reversing transform | |
| *recover_svd* | ☐ | Recover singular values and eigenvectors of XY | |

| ADVANCED | | | GRID? |
|---|---|---|---|
| *score_each_iteration* | ☐ | Whether to score during each iteration of model training. | |
| *max_runtime_secs* | 0 | Maximum allowed runtime in seconds for model training. Use 0 to disable. | ☐ |

Build Model

# Why use PCA?

- Reduce storage space, e.g. 10 GB compressed to 100 MB
- Increase prediction speed, e.g. 10x speed-up with no accuracy loss
- Identify and visualize important features
- Impute missing data entries

- Supports different loss functions:
  - Quadratic (PCA), Absolute, Huber, Poisson, Hinge, Logistic, Periodic
- Supports regularization:
  - Quadratic, L2, L1, NonNegative, OneSparse, UnitOneSparse, Simplex

# More Information

- Anqi Fu GLRM
  - https://www.youtube.com/watch?v=gEZtZRANeLc
- Madeline Udell
  - https://www.youtube.com/watch?v=zwvzGuS82MA
- https://web.stanford.edu/~boyd/papers/pdf/glrm.pdf


avkash@h2o.ai