1. For mixed data type, GBM is the best choice.
2. If there are too many missing value, consider dropping the feature. "too many" means 80% of missing values. 65% is also a good critical point. Another way to is to convert feature of too many missing values into a 0/1 binary feature (0 means missing value, 1 means with value).
3. Find golden features by yourself.
4. Deep learning is good for homogeneous data type.
5. For algorithms that require homogeneous data type, use one-hot encoding to transform categorical data into numerical data, or use categorical PCA to automatically convert into positive integers. The PCA also reduces dimensions.
6. ZIP code is hierarchical categorical data.
7. Date/Time can be converted into features like "Holiday or not", "winter or summer".
8. For ordinal data like FICO, normally use step of 10 or 15 to bin it.
9. There is no automatic or intelligent binning process for any data. Always conduct manual inspection and binning by looking at histogram of the feature. The NumPy package has a function to digitize a feature, but H2o does not have such thing.
10. Be careful about binning because it may cause data leakage if you use simple binning method or introducing target values. An example of data leakage is that when you convert date/time with different method on different data source.
11. Typical binning methods are equal size or equal frequency. Normally 10 to 20 bins.
12. For time-based data, the validation scheme is critical. Scheme 1 is to train using previous data and validate using recent time data. Scheme 2 is to train and validate using random sampled data.
13. Naive Bayes model is good for text processing such as spam detection. This model has a strong assumption on data features (fully independent across all features).
14. H2o's R package has ensemble (stacking) on the fly.
15. When performing ensemble, use totally different models that make different mistakes. Favorite are GBM + Linear (linear regression, logistic regression, GLM).
16. Also for ensemble, make sure to use different features and different encoding for different models. Say, one-hot encoding for categorical data for Model 1, frequency encoding for categorical data for Model 2. Then the ensemble will perform good.
17. Ensemble for time-sensitive data, may consider building fast (short term) models and slow (long term) models. Also consider building "common season" models and "holiday season" models.
18. To programmably use H2o, one can program in R or Python. R package is most flexible. Scala is also used by large companies.
19. For unbalanced dataset, oversampling may create noises if the responder observations are not the universe of responders.
20. For unbalanced data, AUC and lift of top are good choice of stopping metric.
21. Appendix A is an excellent doc for tuning GBM. Important parameters are number of trees (1000), learning rate, depth of tree (10). Favorite is the nbins which binning feature before splitting the tree, it dynamically reduce branches. Also the weights of columns is a parameter to be tuned.
22. For say, 500 combinations of parameters, to programmably run through them and find the best combination, there is a functionality called "grid search" implemented in H2o. It search through the grid of hyper parameter. No random grid search yet though.
23. The AUC in H2o is ROC AUC. To get PR AUC, some custom function hook is needed.

H2o support team,
Below is the notes that I took from conversation between Dmitry and DM team on 12/27/2016.
I would like you to review the notes and correct wrong understanding, if any.
Most importantly, I would like Dmitry to elaborate on
1. How to convert ZIP code into numerical feature
2. How to convert date into numerical feature
3. How to convert categorical feature into numerical feature based on "frequency"
P.S.

Hi John,

Here are the answers:
1. How to convert ZIP code into numerical feature

Two approaches:
- straight forward: Treat ZIP column as any other categorical features and encode it with frequency (see question number 3)
- ZIP is hierarchical code: From wikipedia: "ZIP Codes are numbered with the first digit representing a certain group of U.S. states, the second and third digits together representing a region in that group (or perhaps a large city) and the fourth and fifth digits representing a group of delivery addresses within that region". So you could split up ZIP code into 3 columns: 1 digit, 2 and 3rd digits, 4th and 5th digits and treat them as separate categorical features

2. How to convert date into numerical feature

Let's say you have a date: 2001-02-03 (YYYY-MM-DD) Here is the list of features that can be used instead:
1) Year - 2001
2) Month - 02
3) Day - 03
4) Days from beginning of the year - 34
5) Day of the week - Friday (you could encode it into integer as well, in case if week starts with Sunday=0, it might be encoded as 5)
6) Week of the year - 5

So instead of 1 column with date we could have 6 columns showing different information extracted from the this date.

3. How to convert categorical feature into numerical feature based on "frequency"

you replace each unique category in the column with value n / N,

where:
n - number of rows with this category id in training dataset
N - number of rows in training dataset

As an example:
In your dataset you have a column with categorical values:

Black
Green
Black
Yellow
Green
Black
Black

You will replace “Black” with 4 / 7 = 0.57
“Yellow” with 1 / 7 = 0.14
“Green” with 2 / 7 = 0.28