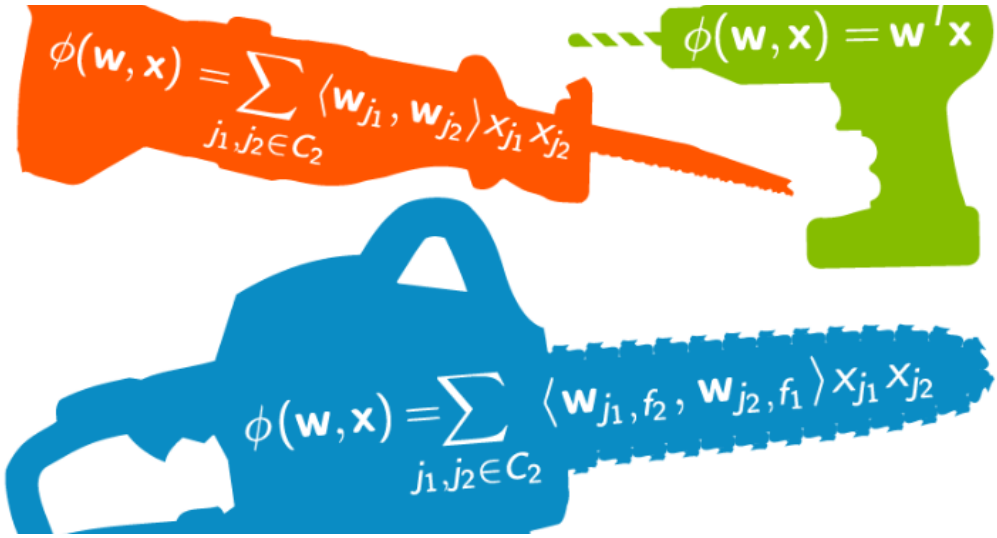


Feature Engineering: Data scientist's Secret Sauce !

Published on July 10, 2016

Ashish Kumar | Follow

It is very tempting for data science practitioners to opt for the best known algorithms for a given problem.*However* It's not the algorithm alone , which can provide the best solution ; Model built on carefully engineered and selected features can provide far better results.

- "Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius -- and a lot of courage -- to move in the opposite direction."- Albert Einstein

The complex models are not easily interpretable and tougher to tune. Simpler algorithms, with better features or more data can perform far better than a weak assumption accompanied with a complex model.

Better features means flexibility, simpler models, better results. Presence of irrelevant features hurt generalization. Thus feature selection and feature engineering should not be considered as mutually exclusive activities and should be performed in conjunction to each other. With the help of an effective feature engineering process, we intend to come up with an effective representation of the data. The question arises, what is considered to be a good or bad representation?

Representation is as good as the information it contains.

Entropy: Higher the entropy, more the information contained in the data , variance: higher the variance: more the information , projection for better separation: the

class association etc , all of these explains the information in data.

Feature engineering is a vital component of modelling process, and it is the toughest to automate. It takes domain expertise and a lot of exploratory analysis on the data to engineer features

1. single variable Basic transformations: x, x^2 ,sqrt x ,log x, scaling
2. If variable's distribution has a long tail, apply Box-Cox transformation (taking log() is a quick & dirty way).
3. One could also perform analysis of residuals or log-odds (for linear model) to check for strong nonlinearities.
4. Create a feature which captures the frequency of the occurrence of each level of the categorical variable. For high cardinality, this helps a lot. One might use ratio/percentage of a particular level to all the levels present.
5. For every possible value of the variable, estimate the mean of the target variable; use the result as an engineered feature.
6. Encode a variable with the ratio of the target variable.
7. Take the two most important variables and throw in second order interactions between them and the rest of the variables - compare the resulting model to the original linear one
8. if you feel your solutions should be smooth, you can apply a radial basis function kernel . This is like applying a smoothing transform.
9. If you feel you need covariates , you can apply a polynomial kernel, or add the covariates explicitly
10. High cardinality features : convert to numeric by preprocessing: out-of-fold average.
    ----T**wo variable combinations**---
11. Additive transformation
12. difference relative to baseline
13. Multiplicative transformation : interactive effects
14. divisive : scaling/normalisation
15. thresholding numerical features to get boolean values
16. Cartesian Product Transformation
17. Feature crosses: cross product of all features.

Consider a feature A, with two possible values {A1, A2}. Let B be a feature with possibilities {B1, B2}. Then, a feature-cross between A & B (let’s call it AB) would

basically give these "combinations" any names you like. Just remember that every combination denotes a synergy between the information contained by the corresponding values of A and B.

18. Normalization Transformation:

- One of the implicit assumptions often made in machine learning algorithms (and somewhat explicitly in Naive Bayes) is that the the features follow a normal distribution. However, sometimes we may find that the features are not following a normal distribution but a log normal distribution instead. One of the common things to do in this situation is to take the log of the feature values (that exhibit log normal distribution) so that it exhibits a normal distribution.If the algorithm being used is making the implicit/explicit assumption of the features being normally distributed, then such a transformation of a log-normally distributed feature to a normally distributed feature can help improve the performance of that algorithm.

19. Quantile Binning Transformation

20. whitening the data

21. Windowing • If points are distributed in time axis, previous points in the same window are often very informative

22. Min-max normalization : does not necessarily preserve order

23.sigmoid / tanh / log transformations

24. Handling zeros distinctly – potentially important for Count based features

25. Decorrelate / transform variables

26. Reframe Numerical Quantities

27. Map infrequent categorical variables to a new/separate category.

28.Sequentially apply a list of transforms.

29. One Hot Encoding

30. Target rate encoding

31. Hash Trick

Multivariate:

32. PCA

33. MODEL STACKING

34. compressed sensing

35.guess the average" or "guess the average segmented by variable X"

----projection : new basis--

36.hack projection:

- Perform clustering and use distance between points to the cluster center as a feature
- **PCA/SVD**

Useful technique to analyze the interrelationships between variables and perform dimensionality reduction with minimum loss of information.

* Find the axis through the data with highest variance.
* Repeat with the next orthogonal axis and so on , until you run out of data or dimensions. Each axis acts a new feature

37.sparse coding

choose basis : Evaluate the basis based on how well you can use it to reconstruct the input and how sparse it is;take some sort of gradient step to improve that evaluation.

- efficient sparse coding algorithms
- deep auto encoders

38 :Random forest: train a bunch of decision trees :use each leaf as a feature

Tagged in: machine learning, feature extraction, data science

Ashish Kumar
10 articles

Follow

10 comments Newest

Sign in to leave your comment

Manish Nair
MSEE, Research Postgrad
··· 7mo

As a frequent user of numerical methods (for electromagnetic simulations), I found this handy. Close to a handbook (almost). Thanks

Like Reply

Data Scientist chez Chauffeur Privé

Great Article !

Like Reply | 1

There are 8 other comments. Show more.

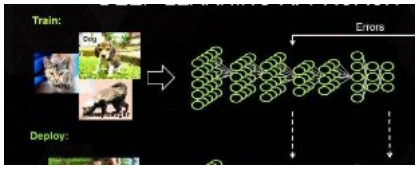
Don't miss more articles by Ashish Kumar

Machine learning as a service ? Might lose sleep over this !

Ashish Kumar on LinkedIn

Machine Learning : Few rarely shared trade secrets

Ashish Kumar on LinkedIn



Deep learning & XgBoost : Winning i hands down !

Ashish Kumar on LinkedIn

Looking for more of the latest headlines on LinkedIn?

Discover more stories