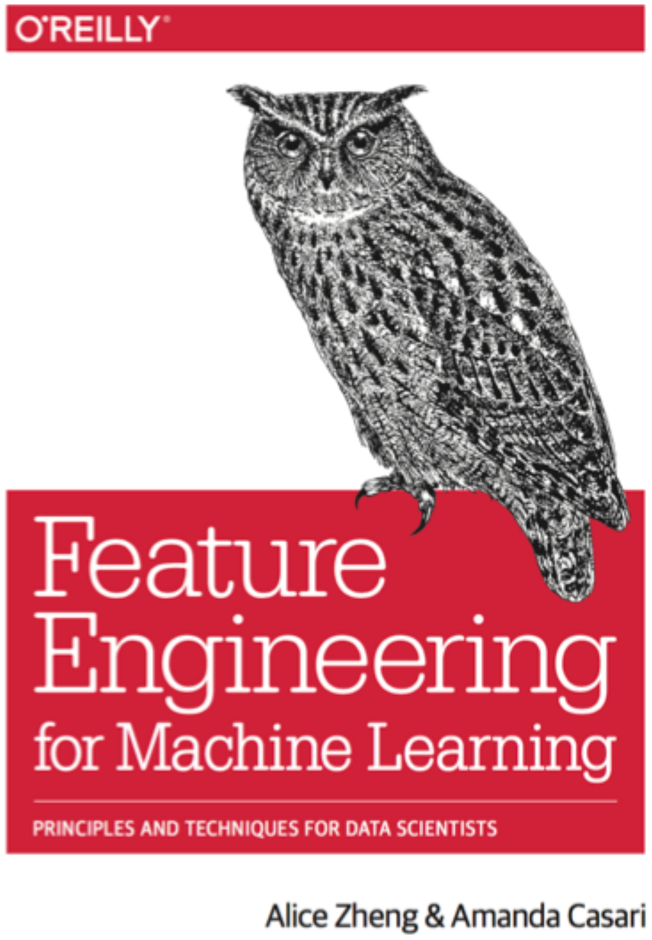
FEATURE ENGINEERING FOR MACHINE LEARNING

William Steimel B1778102

SOURCE



O'REILLY®

Feature Engineering for Machine Learning

PRINCIPLES AND TECHNIQUES FOR DATA SCIENTISTS

Alice Zheng & Amanda Casari

MOTIVATION

Feature Engineering is a data Pre-processing task that can improve predictive performance greatly

Feature Engineering often helps people create the best models and win Kaggle Competitions

I want to use this presentation as my own reference of Feature Engineering techniques

TABLE OF CONTENTS

I. MACHINE LEARNING PROCESS

MACHINE LEARNING PROCESS

A *feature* is a numeric representation of raw data.

Feature engineering- "the process of formulating the most appropriate features given the data, the model, and the task."

Today's presentation will largely talk about Feature Engineering techniques based on pre-processing data and model based methods

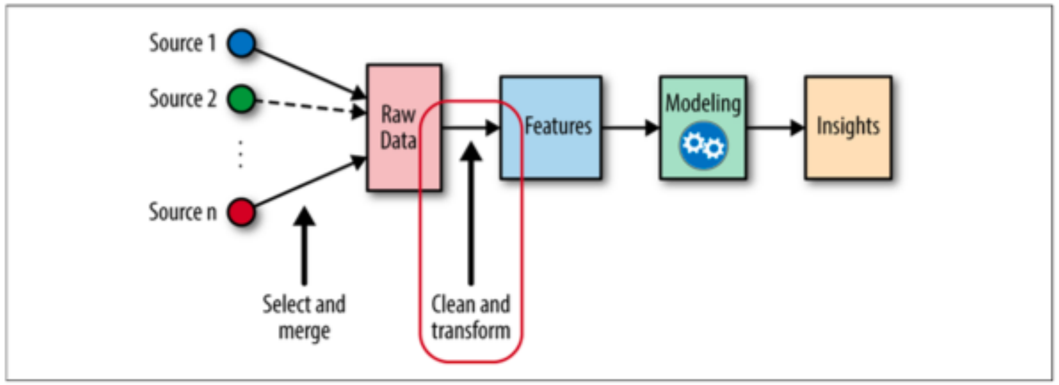


Figure 1-2. The place of feature engineering in the machine learning workflow

II. FANCY TRICKS WITH SIMPLE NUMBERS

Techniques for Numeric Data

FANCY TRICKS WITH SIMPLE NUMBERS

This section starts with the simplest data: Numeric Data

Examples of Numeric Data:

- Geolocation of a person
- Measurements from sensor
- Traffic counts, etc.

A few considerations are important before Feature Engineering with numeric data:

- Does Magnitude Matter? Do we need to know whether its positive or negative?
- Scale of features? Do the largest and smallest values span several orders of magnitude?
  - Some algorithms that depend on Euclidean Distance like k-means clustering are sensitive to scaling
- Distribution – The data distribution of features can influence some models more than others
  - Linear Regression assumes a gaussian distribution

FANCY TRICKS WITH SIMPLE NUMBERS

When dealing with counts it is important to:

- Determine scale and whether to keep data as raw numbers
- Conversion into binary values to indicate presence
- Create bins or partitions for coarser granularity

This section will cover these techniques

BINARIZATION

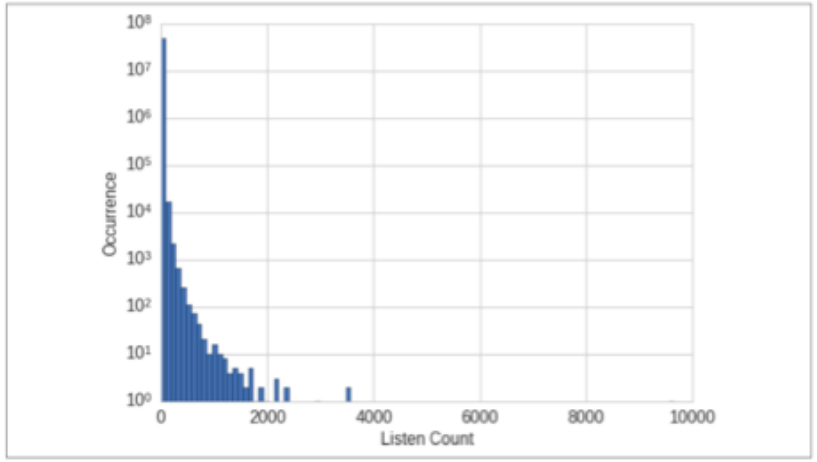


Figure 2-3. Histogram of listen counts in the Taste Profile subset of the Million Song Dataset—note that the y-axis is on a log scale

Binarization is an indication of presence (1 if yes, 0 if no)

The Echo Nest Taste Profile subset is a dataset that contains the full music listening histories of one million users on Echo Nest

- Song Listen Counts
- The goal is to predict whether a user might like a song based on their listen count

There are many different listen counts including one user who listened to a song 9,667 times and some users who have only listened to a song once

- Raw listen count is not an effective measure of user taste
- Some users listen to songs only once and sometimes many times on loop

One way to handle this problem would be to binarize each count greater than or equal to 1 to be 1

- A user who listens to a song at least once can be assumed to like the song

QUANTIZATION OR BINNING

Binning is the splitting or grouping of numerical data into groups

This section uses the Yelp dataset challenge which contains user reviews of businesses from 10 cities across North America and Europe

- Each business is labeled with 0 or more categories

The business objective is to predict what rating a user might give to a business

- Review count may be good as review count is often an indicator of popularity
- However, raw counts that span several orders of magnitude (High Variance) are challenging for many models

One solution is quantizing the counts or grouping the counts into bins

There are two types of binning and one needs to decide how wide each bin should be when quantizing data

- Fixed-width binning
- Quantile binning

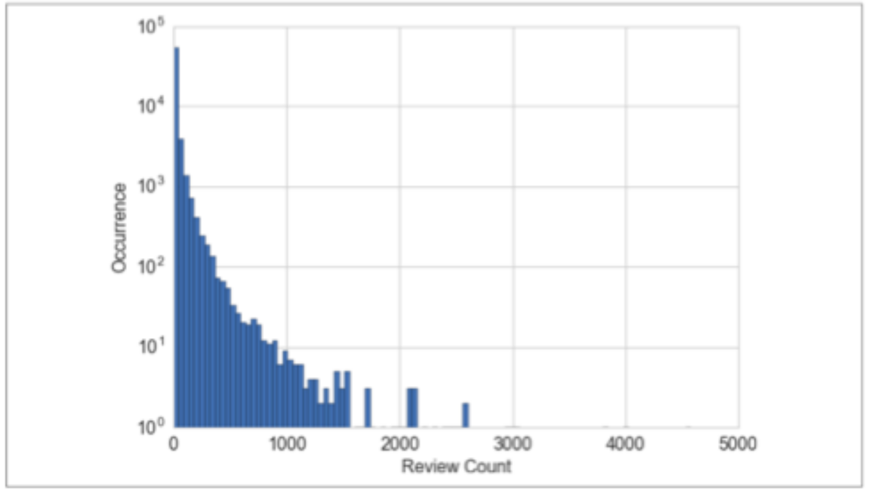
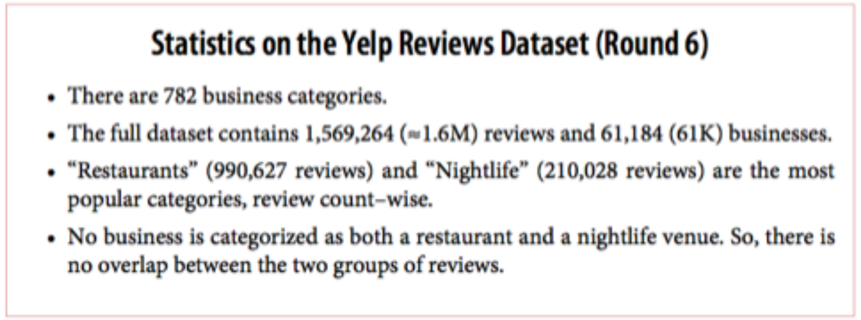


Figure 2-4. Histogram of business review counts in the Yelp reviews dataset—the y-axis is on a log scale



Statistics on the Yelp Reviews Dataset (Round 6)

- There are 782 business categories.
- The full dataset contains 1,569,264 (≈1.6M) reviews and 61,184 (61K) businesses.
- "Restaurants" (990,627 reviews) and "Nightlife" (210,028 reviews) are the most popular categories, review count–wise.
- No business is categorized as both a restaurant and a nightlife venue. So, there is no overlap between the two groups of reviews.

QUANTIZATION OR BINNING

Fixed-Width Binning – Each bin contains a specific numeric range

- Age Example
  - 0-12 years old
  - 12-17 years old
  - 18-14 years old
  - 15-34 years old
  - 35-44 years old
  - 45-54 years old
  - 55-64 years old

Quantile Binning – Adaptive binning based on quantiles of the data distribution

- Quantiles are values that divide the data into equal parts
  - Median – divides the data in half
  - Quartiles – divides the data in quarters
  - Deciles – divides the data in tenths

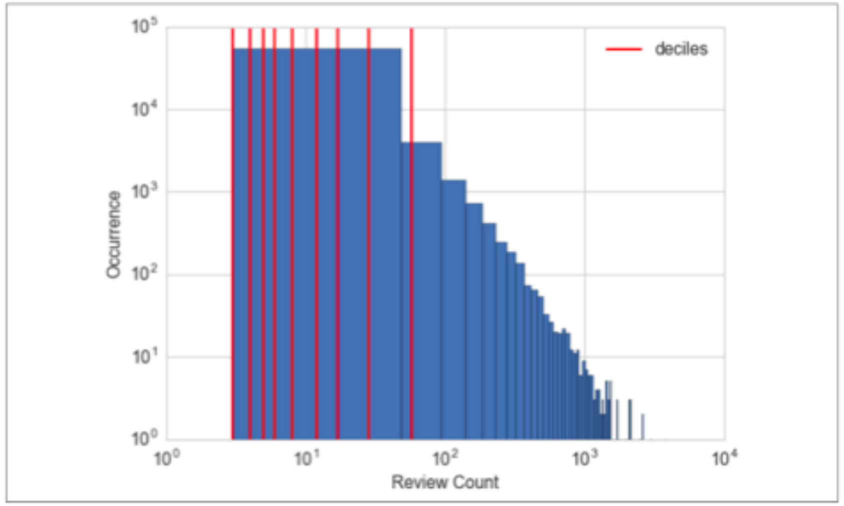


Figure 2-5. Deciles of the review counts in the Yelp reviews dataset—both the x- and y-axes are on a log scale

LOG TRANSFORMATION

The log function is the inverse of the exponential function

The Log transform is useful for dealing with positive numbers with a heavy tail distribution

- "compresses the range of small numbers."

Highly skewed datasets become less skewed and more spread over the x-axis

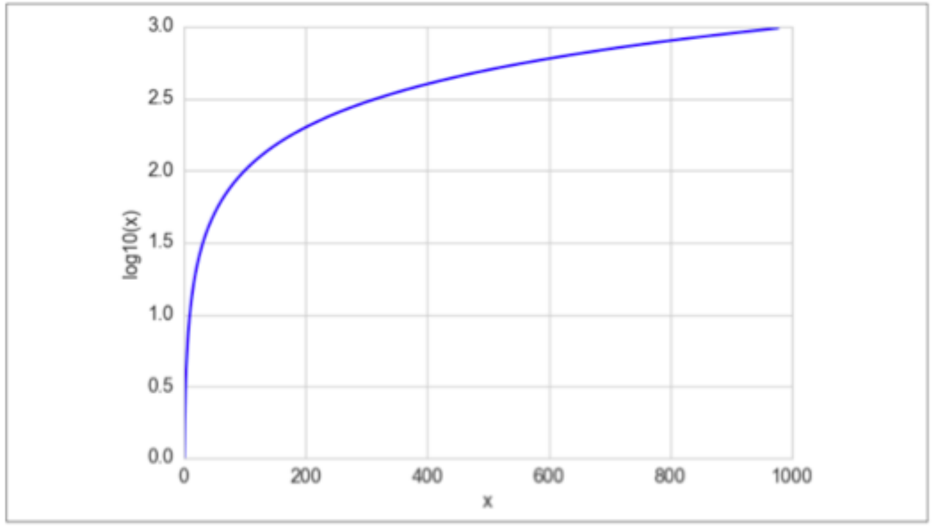


Figure 2-6. The log function compresses the high numeric range and expands the low range

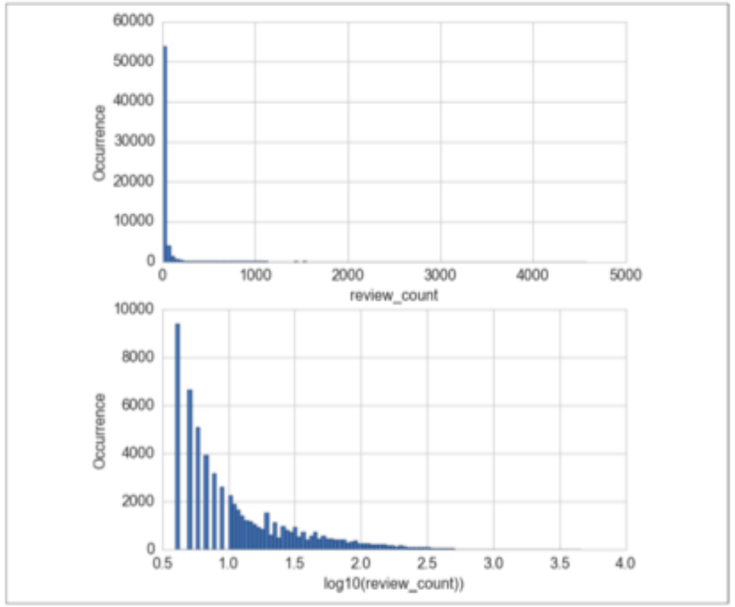


Figure 2-7. Comparison of Yelp business review counts before (top) and after (bottom) log transformation

POWER TRANSFORMS: GENERALIZATION OF THE LOG TRANSFORM

The Log Transform belongs to a family of transformations called Power transforms

- variance-stabilizing transformations

Power Transforms change the distribution of variables so that the variance is no longer dependent on the mean.

- Another power transform is called the Box-Cox Transform

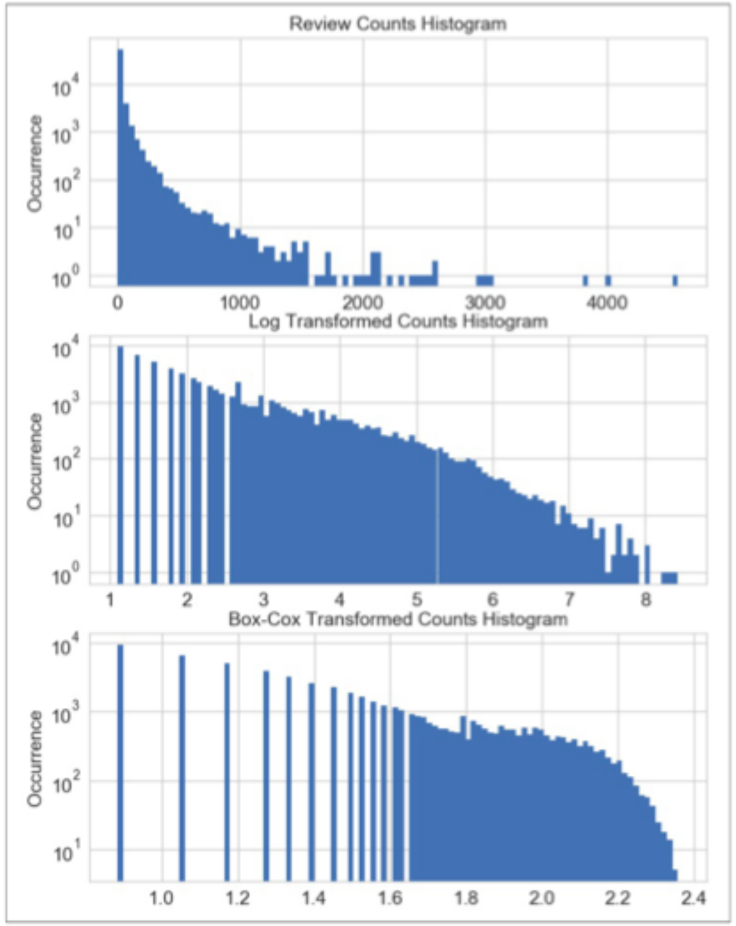


Figure 2-13. Box-Cox transformation of Yelp business review counts (bottom), compared to original (top) and log transformed (middle) histograms

FEATURE SCALING OR NORMALIZATION

"Models that are smooth functions of the input, such as linear regression, logistic regression, or anything that involves a matrix, are affected by the scale of the input."

Feature Scaling can be useful in these situations and changes the scale of features

- Min-Max Scaling
- Standardization
- L2 Normalization

Feature Scaling does not change the shape of the feature distribution

Feature scaling is also useful in situations where features differ in scale greatly

FEATURE SCALING OR NORMALIZATION

Min-Max Scaling- Scales values between 0 and 1

Standardization (Variance Scaling) - Scales to mean of 0 and Variance of 1

L2 Normalization – Divides the original feature value by the l2 norm or Euclidean norm

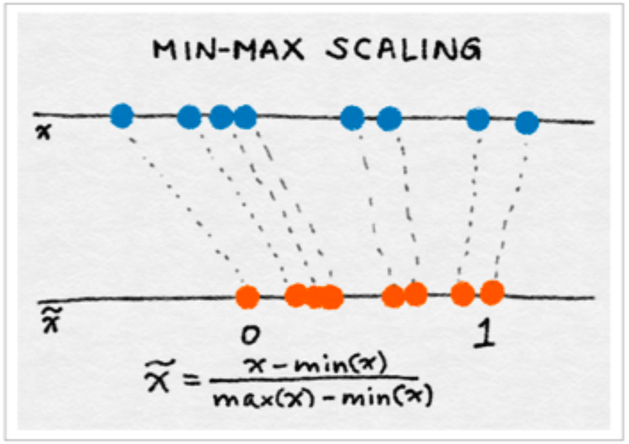


Figure 2-15. Illustration of min-max scaling

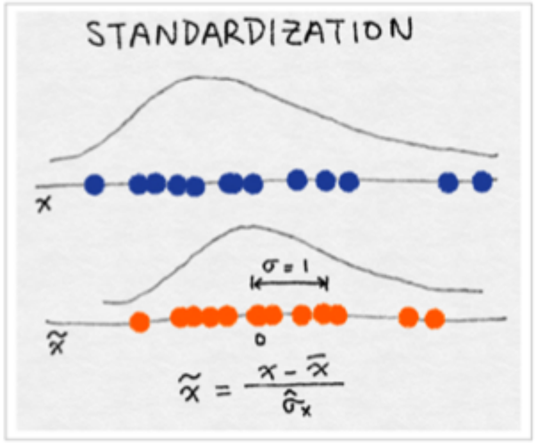


Figure 2-16. Illustration of feature standardization

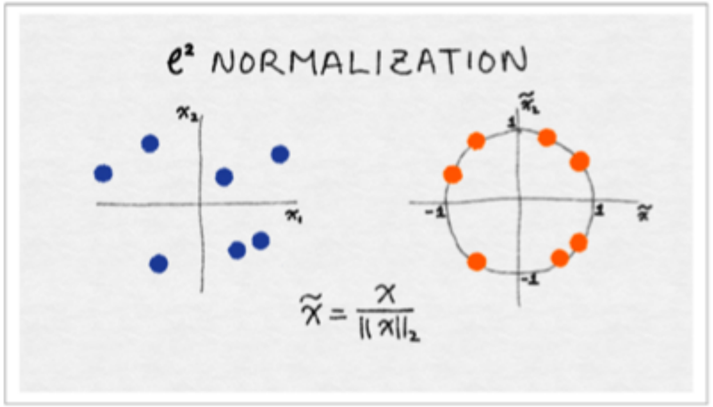


Figure 2-17. Illustration of ℓ^2 feature normalization

INTERACTION FEATURES

Interaction feature – the product of two features

Interaction Features function like the Logical AND:

- Example: Zip code 98121 AND the user's age is between 18 and 35

Generalized linear models often see improvements with interaction features

Generally a linear model wishes to predict the outcome y based on individual input features x_1, x_2,… x_n

<u>Linear Model</u>

$$y = w_1 x_1 + w_x x_2 + \cdots + w_n x_n$$

<u>Linear Model with Interaction features</u>

$$y = w_1 x_1 + w_x x_2 + \cdots + w_n x_n + w_{1,1} x_1 \ x_1 + w_{1,2} x_1 \ x_2 + w_{1,3} x_1 \ x_3 + \cdots$$

III. CATEGORICAL VARIABLES: COUNTING EGGS IN THE AGE OF ROBOTIC CHICKENS

Techniques for Categorical Variables

CATEGORICAL VARIABLES: COUNTING EGGS IN THE AGE OF ROBOTIC CHICKENS

A Categorical variable is used to represent categories or labels

- Major cities in the world
- The four seasons in a year
- Industry of a company

Categorical Variables are considered nonordinal as they cannot be ordered in respect with one another

This section also discusses large Categorical variables which are common in transactional records

- ID
- IP address

ENCODING CATEGORICAL VARIABLES

Categorical variables like eye color can be black, blue, brown but an encoding method is needed to turn these nonnumeric categories into numbers

Some methods for encoding categorical variables include:

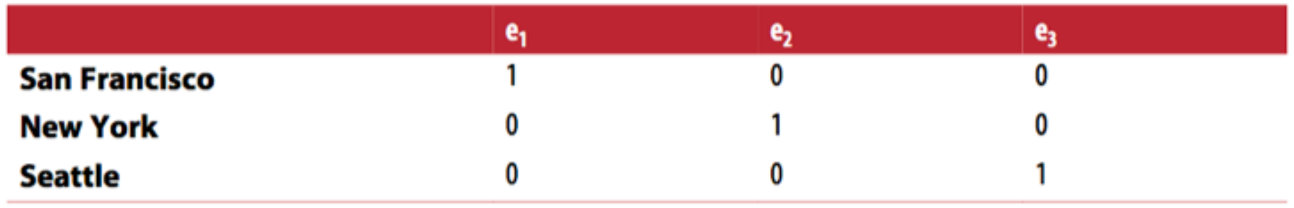
- One-Hot Encoding
- Dummy Coding
- Effect Coding

ONE-HOT ENCODING

One-Hot Encoding - Each bit represents a possible category and only one bit can be "on."

A categorical variable with k possible categories is encoded as a feature vector of length k

Table 5-1. One-hot encoding of a category of three cities



	e_1	e_2	e_3
San Francisco	1	0	0
New York	0	1	0
Seattle	0	0	1

DUMMY CODING

Dummy Coding removes extra degree of freedom by using only k-1 features in the representation.

One feature is represented by the vector of all zeros ("Reference Category")

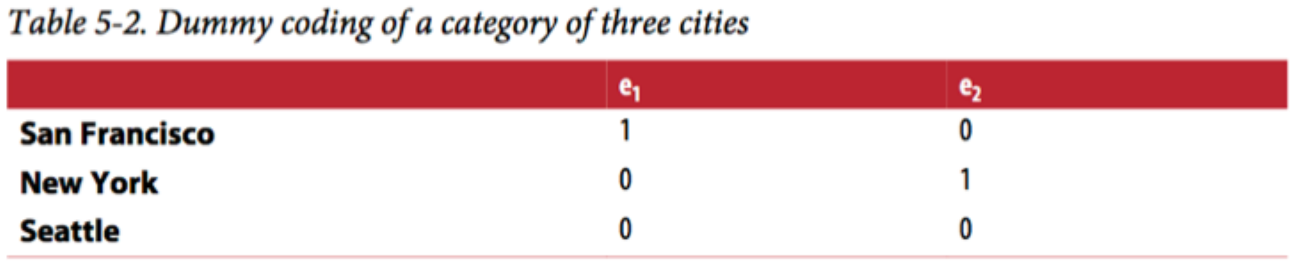


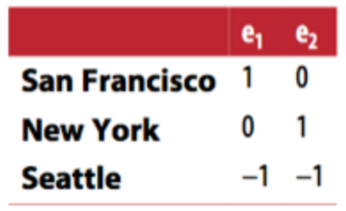
Table 5-2. Dummy coding of a category of three cities

	e_1	e_2
San Francisco	1	0
New York	0	1
Seattle	0	0

EFFECT CODING

Effect Coding is another variant of categorical variable encoding.

Similar to dummy encoding but the reference category is instead represented by a vector of all -1's



	e_1	e_2
San Francisco	1	0
New York	0	1
Seattle	–1	–1

COMPARISON BETWEEN CATEGORICAL VARIABLE ENCODINGS

All methods are very similar

One-hot encoding is redundant which means multiple valid models for the same problem

Dummy Coding and effect coding are not redundant and allow for unique and interpretable models

Effect coding is a dense vector of all -1's which is expensive for both storage and computation

- Pandas and scikit-learn avoid this and use dummy coding and one-hot encoding by default

All three encoding techniques are not very good with a huge number of categories

DEALING WITH LARGE CATEGORICAL VARIABLES

Automated data collection has lead to datasets with large categorical variables

- Targeted Advertising
- Fraud Detection

A key challenge is finding a good feature representation that produces models that are memory efficient, fast, and accurate.

Solutions to handle this problem:

- 1. Proceed as normal - Use a simple model and feed one-hot encoding into linear model on lots of machines (Microsoft Search Advertising Engine)
- 2. Compress the features
  - Feature hashing (Yahoo!)
  - Bin Counting (Microsoft)

FEATURE HASHING

Hash function – "a deterministic function that maps a potentially unbounded integer to a finite integer range [1, *m*]

- Collision - Multiple numbers may get mapped to the same input

We can think of the hash function as a machine that takes numbered balls (keys) and routes them to one of m bins

- This maintains feature space while reducing storage and processing time during Machine Learning
- Hashing sacrifices user interpretability of features

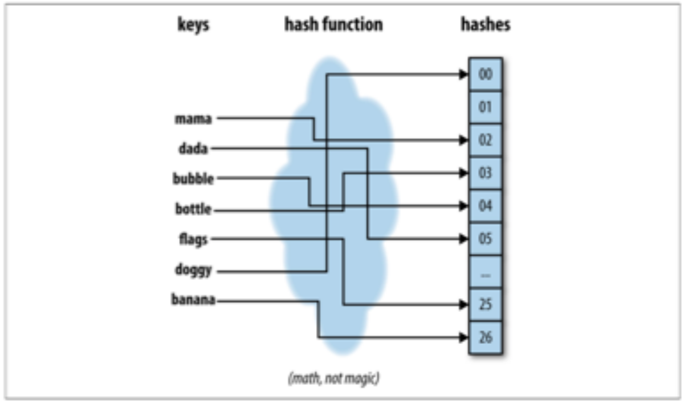


Figure 5-1. Hash functions map keys to bins

BIN COUNTING

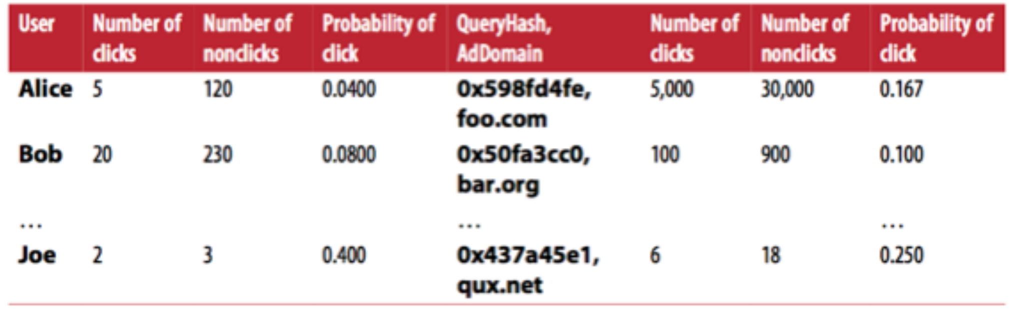
Bin Counting – Instead of using the value of the categorical feature as a feature it instead uses the conditional probability of the target under that value

- Converts categorical variable into statistical values
- Example – Based on the number of times Alice has clicked and not clicked we can calculate the probability of clicking an ad
- Other features including historical click-through probability, raw number of clicks/nonclicks, log-odds ratio, and any other measures of probability can be included.

Used in a variety of applications

- Ad click-through rate prediction
- Hardware branch prediction

Table 5-6. Example of bin-counting features (reproduced from "Big Learning Made Easy—with Counts!" with permission)



User	Number of clicks	Number of nonclicks	Probability of click	QueryHash, AdDomain	Number of clicks	Number of nonclicks	Probability of click
Alice	5	120	0.0400	**0x598fd4fe, foo.com**	5,000	30,000	0.167
Bob	20	230	0.0800	**0x50fa3cc0, bar.org**	100	900	0.100
...				...			...
Joe	2	3	0.400	**0x437a45e1, qux.net**	6	18	0.250

WHAT ABOUT RARE CATEGORIES?

Rare Categories require special treatment

- Users who log in once a year will have very little data to estimate their click through rate
- Rare categories can be a waste of space in the counts table

- A technique called back-off or count-min sketch can be used

WHAT ABOUT RARE CATEGORIES?

- Back-off – "a simple technique that accumulates the counts of all rare categories in a special bin"
  - If counts are greater than a threshold than the category gets its own count statistics otherwise statistics from the back-off bin are used

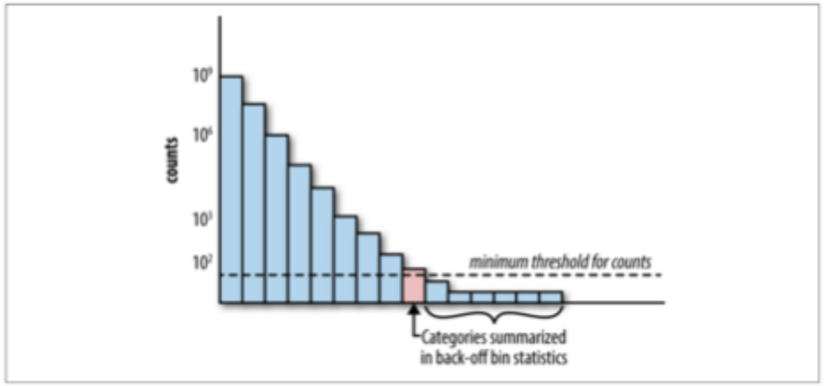


Figure 5-3. If a rare category gains counts, it can move above the threshold for the back-off bin, using its own count statistics for modeling

WHAT ABOUT RARE CATEGORIES?

Count-min sketch- all categories are mapped to frequencies through multiple hash functions with an output range m, smaller than the number of categories k

Figure 5-4 illustrates. Each item i is mapped to one cell in each row of the array of counts. When an update of c_t to item i_t arrives, c_t is added to each of these cells, hashed using functions $h_1 \ldots h_d$.

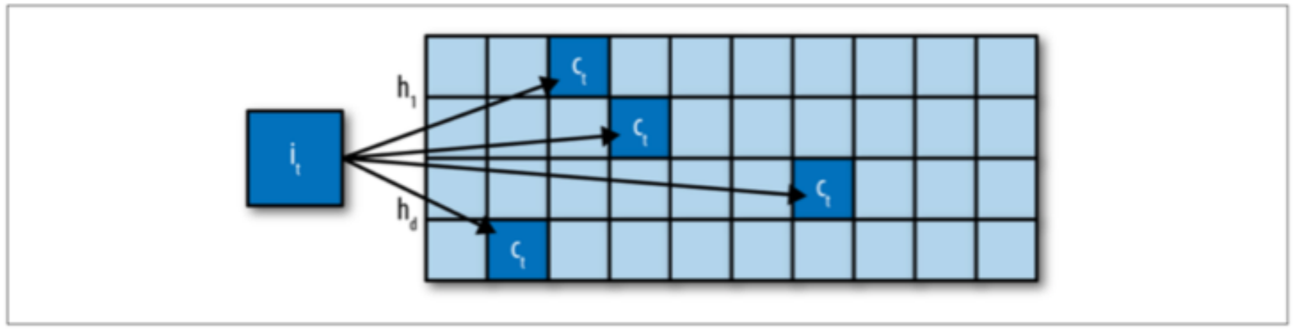
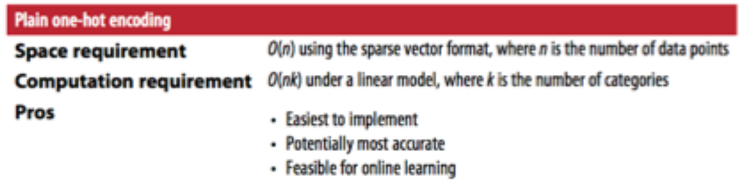


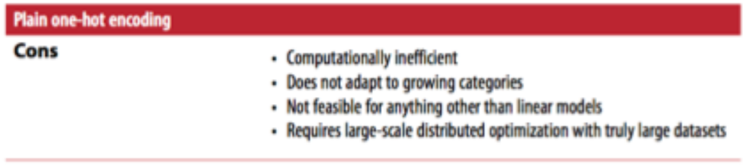
Figure 5-4. The count-min sketch

SUMMARY OF CATEGORICAL FEATURE HANDLING

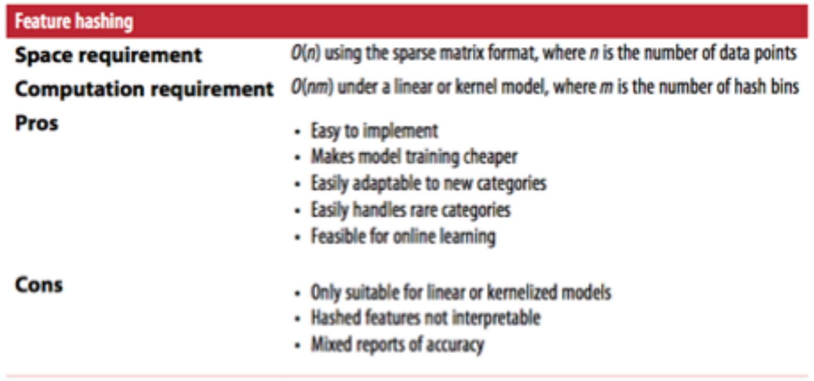
No method is perfect and depends on the desired model



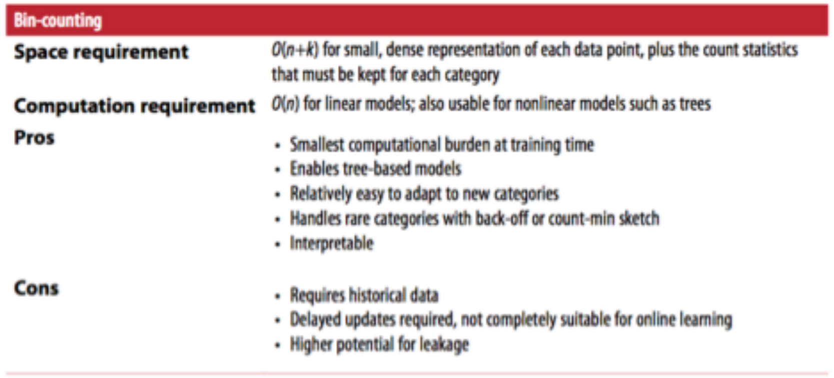
Plain one-hot encoding	
Space requirement	$O(n)$ using the sparse vector format, where n is the number of data points
Computation requirement	$O(nk)$ under a linear model, where k is the number of categories
Pros	• Easiest to implement • Potentially most accurate • Feasible for online learning



Plain one-hot encoding	
Cons	• Computationally inefficient • Does not adapt to growing categories • Not feasible for anything other than linear models • Requires large-scale distributed optimization with truly large datasets



Feature hashing	
Space requirement	$O(n)$ using the sparse matrix format, where n is the number of data points
Computation requirement	$O(nm)$ under a linear or kernel model, where m is the number of hash bins
Pros	• Easy to implement • Makes model training cheaper • Easily adaptable to new categories • Easily handles rare categories • Feasible for online learning
Cons	• Only suitable for linear or kernelized models • Hashed features not interpretable • Mixed reports of accuracy



Bin-counting	
Space requirement	$O(n+k)$ for small, dense representation of each data point, plus the count statistics that must be kept for each category
Computation requirement	$O(n)$ for linear models; also usable for nonlinear models such as trees
Pros	• Smallest computational burden at training time • Enables tree-based models • Relatively easy to adapt to new categories • Handles rare categories with back-off or count-min sketch • Interpretable
Cons	• Requires historical data • Delayed updates required, not completely suitable for online learning • Higher potential for leakage

IV. Dimensionality Reduction: Squashing the Data Pancake with PCA

Model Based Feature Engineering

DIMENSIONALITY REDUCTION: SQUASHING THE DATA PANCAKE WITH PCA

This section focuses on Feature Dimensionality Reduction using principal component analysis (PCA)

- Model based feature engineering technique
- Model based techniques require information from the data

Dimensionality Reduction’s goal is to get rid of uninformative information and maintain important information

PCA INTUITION

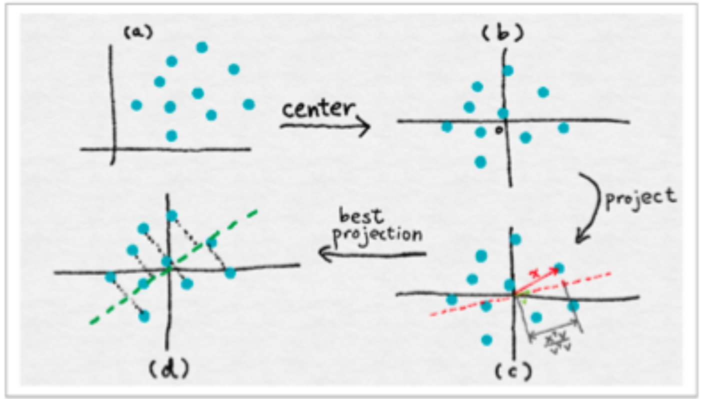


Figure 6-2. Illustration of PCA: (a) original data in feature space; (b) centered data; (c) projecting a data vector x onto another vector v; (d) direction of maximum variance of the projected coordinates (equal to the principal eigenvector of X^TX)

PCA focuses on the notion of linear dependency

- If the column space is small compared to the total number of features then most of the features are linear combinations of a few key features
- PCA tries to reduce this waste of space by squashing data into lower dimensional subspace

"The key idea here is to *replace redundant features with a few new features that adequately summarize information contained in the original feature space*."

Mathematically the goal is to maximize variance of the data points in new feature space

I will cover PCA more in detail in next weeks presentation as I am currently reviewing a paper.

PCA IN ACTION

The Author applies PCA to 100 observations in the MNIST dataset

- (8x8 images) = 64 dimensions
- After PCA the dataset is reduced to 3 dimensions.
  - 3 principal components which contain the most variance

The first three principal components account for 40 % of the total variance (information retained)

PCA is able to group similar numbers together despite there only being 40 % of the original data variance.

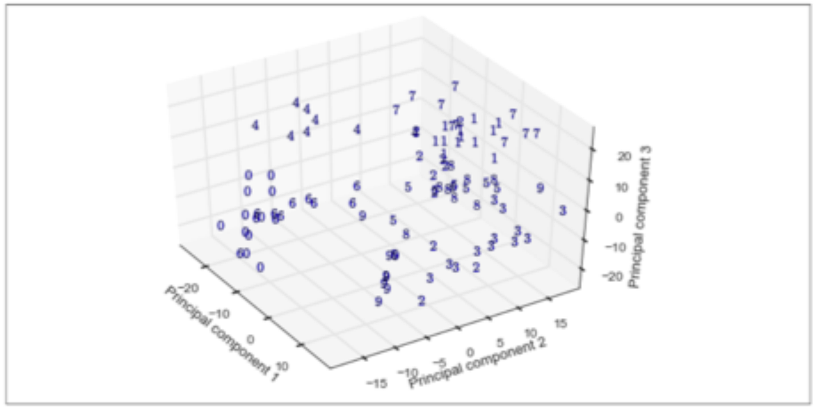


Figure 6-3. PCA projections of subset of MNIST data—markers correspond to image labels

USE CASES - PCA

Anomaly Detection of Time Series Data – PCA was used to diagnose anomalies in internet traffic as the principal components represented the normal traffic trends in the network while the rest of the components represented other signals or anomalies

Financial Modeling – Used as a factor analysis method which is a family of statistical methods that describe variability in data using a small number of unobserved factors

- PCA can be used to find commonly covarying stocks or unexpected correlations

PCA is also often used as a pre-processing step to speed up algorithms

SUMMARY – PCA

Two things to remember about PCA – (Think of it as squashing data into a pancake that is as fluffy (higher variance) as possible)

- Mechanism – Linear Projection
- Objective – Maximize variance of projected data

PCA is a form of Model Driven Feature Engineering

PCA is a well known dimensionality reduction technique but has some limitations

- High computational cost
- Uninterpretable output

V. Nonlinear Featurization via K-Means Model Stacking

Model Stacking for Feature Engineering

NONLINEAR FEATURIZATION VIA K-MEANS MODEL STACKING

PCA is very useful when the data lies in linear space but what if the data forms a more complicated shape?

- The swiss roll is an example of a nonlinear manifold or more complicated data shape

In this example, Nonlinear Dimensionality Reduction is also called nonlinear embedding or manifold learning

The goal of Feature Engineering is to find the right features for the ML task (spatial representation) which K-means can help us with.

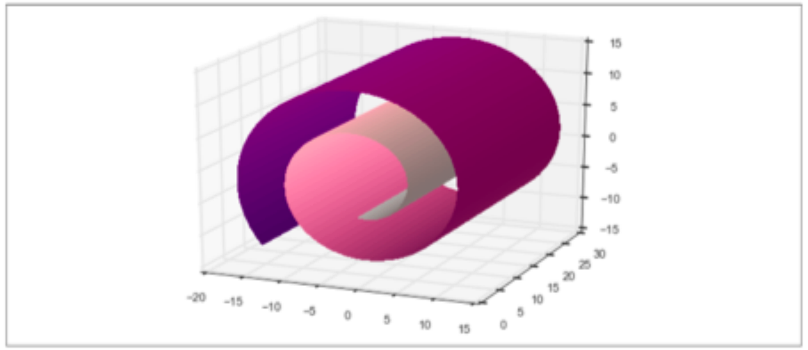


Figure 7-1. The Swiss roll, a nonlinear manifold

K-MEANS

K-means

- Unsupervised Learning
- Clustering Algorithm

K-Means learns to position cluster centers (centroids) so that the total Euclidean distance between each data point and centroid is minimized

Objective Function: $\min_{C_1,\ldots,C_k,\mu_1,\ldots,\mu_k} \sum_{i=1}^{k} \sum_{x\in C_i} \| x - \mu_i \|_2$

The K-means in this example is used for nonlinear manifold feature extraction and is presented as a technique for local structure learning of the swiss roll data

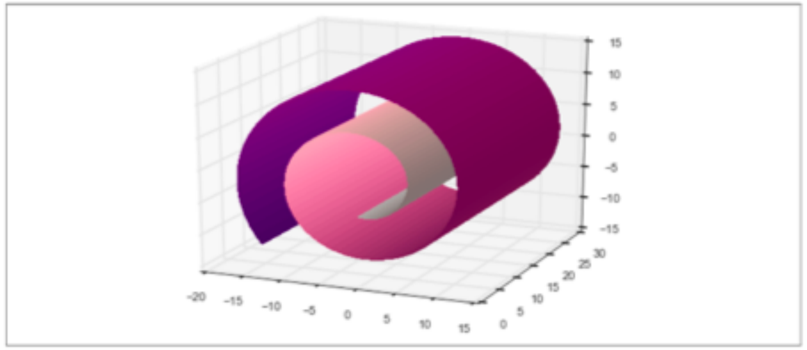


Figure 7-1. The Swiss roll, a nonlinear manifold

Figure 7-2 shows *k*-means at work on two different, randomly generated datasets. The data in (a) is generated from random Gaussian distributions with the same variance but different means. The data in (c) is generated uniformly at random. These toy problems are very simple to solve, and *k*-means does a good job. (The results could be sensitive to the number of clusters, which must be given to the algorithm.)

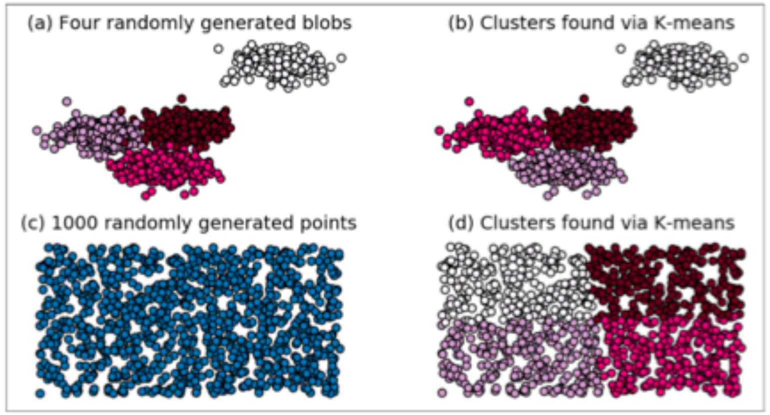


Figure 7-2. k-means examples demonstrating how the clustering algorithm partitions space

CLUSTERING AS SURFACE TILING

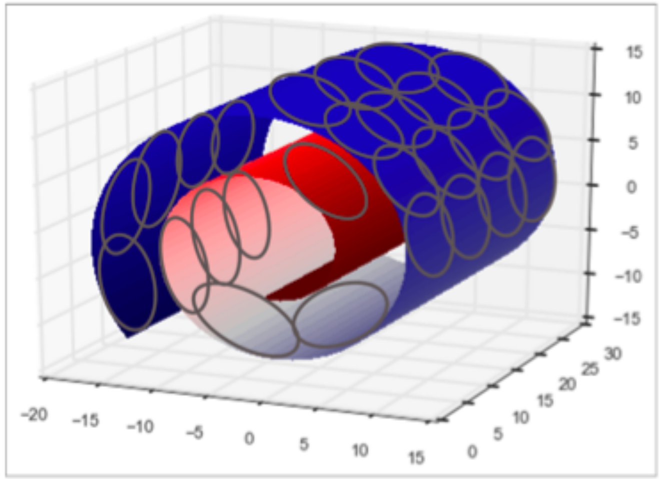


Figure 7-3. Conceptual local patches on the Swiss roll from a clustering algorithm

How can K-means be used to learn local structure?

- Clusters in k-means can be thought of as covering the data surface with patches of clusters

Number of clusters selected greatly influences these patches

- 100 clusters shows a very local result which is good in this example
- 10 clusters clusters together different sections of the manifold (Yellow, Purple, Green, Red)

The output clusters or patches of the K-means can then be used as new features

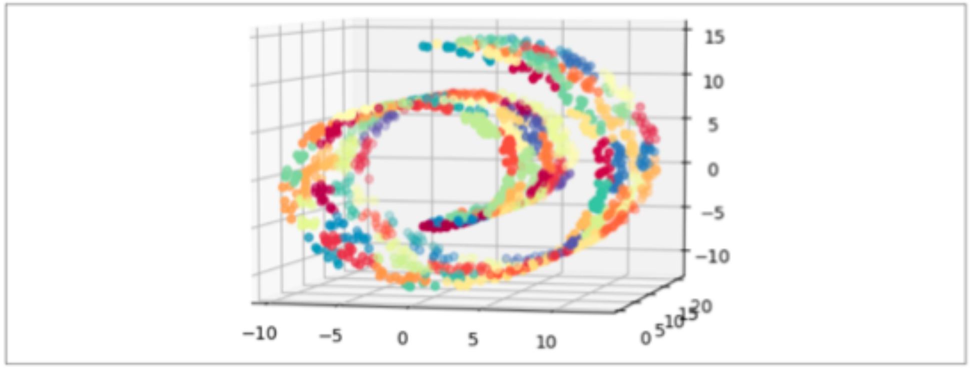


Figure 7-4. Approximating a Swiss roll dataset using k-means with 100 clusters

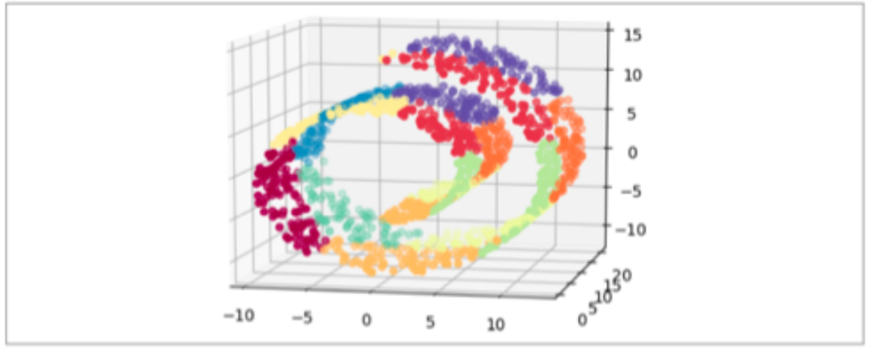


Figure 7-5. k-means on the Swiss roll with 10 clusters

K-MEANS FEATURIZATION FOR CLASSIFICATION

"K-Means Featurization creates a compressed spatial index of the data which can be fed into the model in the next stage."

- This is an example of **model stacking –** output of one model is the input of another.

Spatial data points can be represented as features by its cluster membership in a one-hot encoding vector

Results:

- Logistic Regression with K-Means stacking performed way better than Logistic Regression alone
- Logistic Regression with K-Means is competitive with nonlinear classifiers which is good as linear classifiers are considered less computationally costly

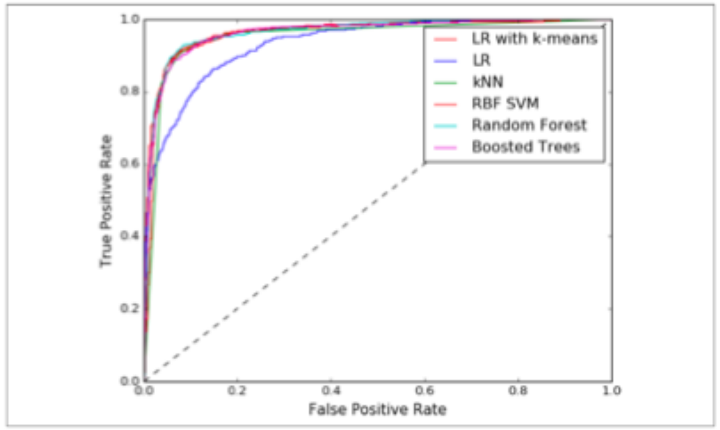


Figure 7-7. ROCs of k-means + logistic regression versus nonlinear classifiers and plain logistic regression on the synthetic two-moons dataset

ALTERNATIVE DENSE FEATURIZATION

Data points can also be represented as a dense vector of inverse distance to each cluster center.

- This retains more information than simple cluster assignment but results in more density.
- This may be more expensive in the modeling step
- There is a tradeoff between sparse and dense features

SUMMARY

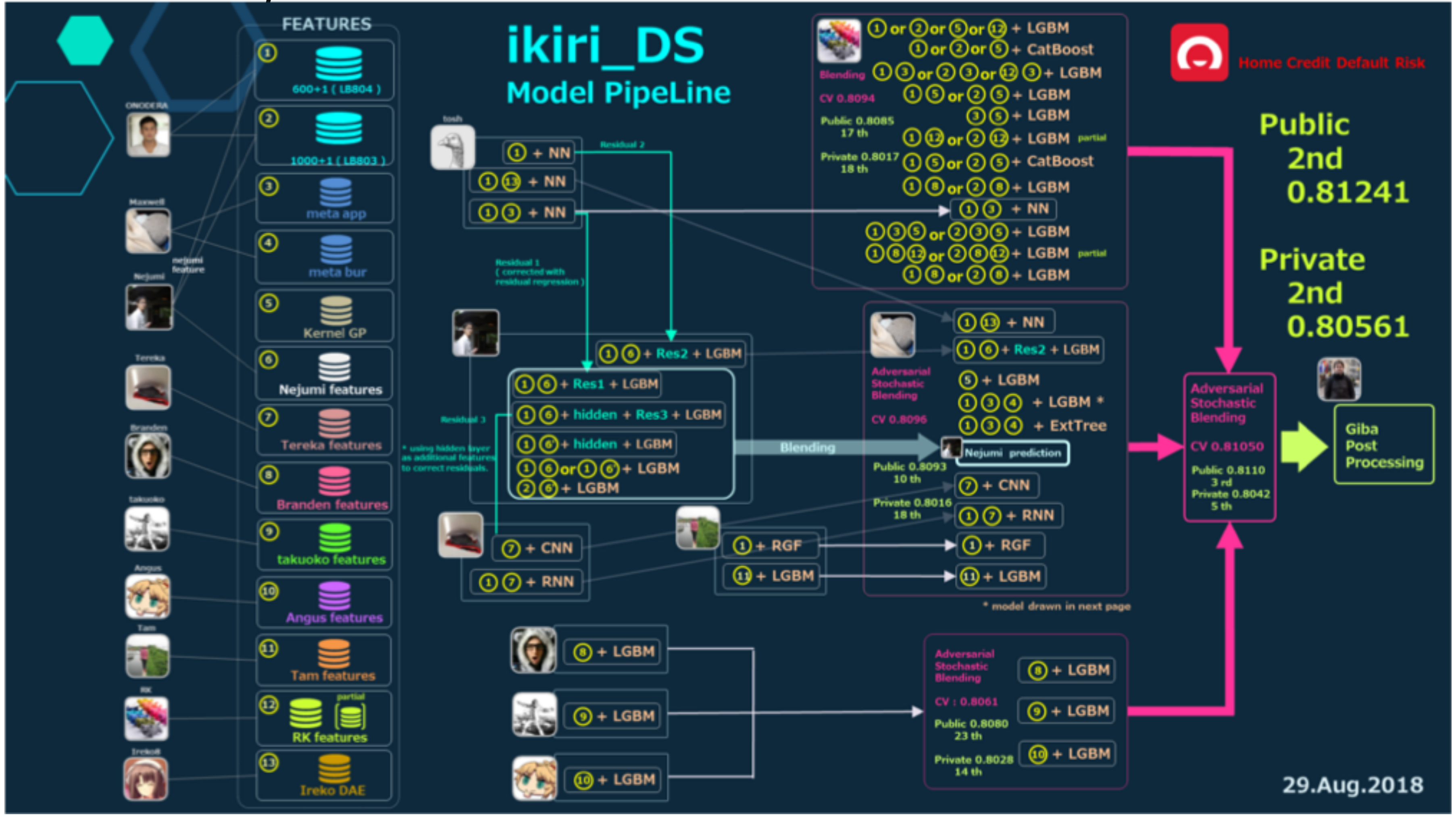
This chapter illustrated the concept of model stacking through the combination of k-means with a simple linear classifier.

Stacking has become an increasingly popular technique recently and the possibilities are endless for combinations

The Author’s Advice – Use stacking to introduce non-linearity into features and use a simple linear model as the last layer

- This creates a balance between performance and accuracy

STACKING/BLENDING EXAMPLE



Source: ikiri_DS Home Credit Default Risk Kaggle Competition

CONCLUSION

This Presentation covered many Feature Engineering Techniques related to numeric data, categorical data as well as model based techniques and Stacking.

Although Feature Engineering for Text Data and Neural Network's have been omitted from this presentation they will be covered in a future presentation.

I look forward to utilizing these techniques in real world problems, research, and Kaggle competitions.

ADDITIONAL RESOURCES

https://www.slideshare.net/HJvanVeen/feature-engineering-72376750