COMP 333 Data Analytics

Exploratory Data Analysis

Greg Butler

Data Science Research Centre

and

Centre for Structural and Functional Genomics

and

Computer Science and Software Engineering
Concordia University, Montreal, Canada

gregb@cs.concordia.ca

Feature Engineering

Feature

A feature is an attribute or property shared by all of the independent units on which analysis or prediction is to be done. Any attribute could be a feature, as long as it is useful to the model.

Process of Feature Engineering

- Brainstorming or Testing features;
- Deciding what features to create;
- Creating features;
- Checking how the features work with your model;
- Improving your features if needed;
- Go back to brainstorming/creating more features until the work is done.

See video 3, Ryan Baker, Coursera, Big Data Week 3 Feature Engineering

https://www.youtube.com/watch?v=drUToKxEAUA

Feature Engineering

FE is a Representation Problem

What is the best representation of the sample data
in order to learn a solution to your problem?

Machine learning algorithms learn a solution to a problem from sample data.

You have to turn your inputs into things the algorithm can understand

FE is an Art

like engineering, like programming, like medicine is an art.

The data is variable and is different every time.

You get good at deciding which procedures to use and when, by practice.

Feature Importance

You can objectively estimate the usefulness of features.

A feature may be important if it is *highly correlated*
with the dependent variable (the thing being predicted).

Correlation coefficients can provide feature importance scores.

Features with the highest scores can be selected
for inclusion in the training dataset,
whereas those remaining can be ignored.

Important features can guide you
to extract or construct new features,
similar but different to those that have been estimated to be useful.

Approaches to Feature Engineering

The key approaches to feature engineering are:

- feature selection
- feature extraction
- feature construction

Feature Learning

Deep learning is feature learning where the set of features is *learnt* from the raw data, by the early layers of the network.

Feature Selection

Feature selection selects a subset of features that are most useful to the problem.

Feature selection algorithms may use a scoring method
to rank and choose features,
such as correlation or other feature importance methods.

More advanced methods may search subsets of features
by trial and error,
creating and evaluating models automatically in pursuit of
the objectively most predictive sub-group of features.

Some machine learning algorithms, such as random forests,
will incorporate feature selection as part of their algorithm.

Feature Extraction

Feature extraction is a process of automatically reducing the dimensionality of these types of observations into a much smaller set that can be modelled.

For tabular data, this might include projection methods like Principal Component Analysis and unsupervised clustering methods.

Examples include
Images into colours, textures, contours, etc
Signals into frequency, phase, samples, spectrum, etc
Time series into ticks, trends, self-similarities, etc
Biomed into dna sequence, genes, etc

Text into words, POS (part-of-speech) tags, grammatical dependencies, etc

Feature Construction

Feature construction is the manual construction of new features from raw data

This requires spending a lot of time with actual sample data (not aggregates) and thinking about
the underlying form of the problem,
the structures in the data, and
how best to expose them to predictive modeling algorithms.

With tabular data, it often means a mixture of
aggregating or combining features to create new features,
and decomposing or splitting features to create new features.

Feature Creation

Aggregation

Basic aggregation operators

- sum
- mean, media, mode
- frequency

Other

- binning

Transformation

Apply a transformation to features

- normalization, unification, resolution, regularization
- log
- feature split
- scaling

Feature Creation: Binning

Numerical Data to Categorical Data

Example: Age

Define **bins**:

`Infant` for *age* between 0 – 4
`Child` for *age* between 5 – 12
`Teen` for *age* between 13 – 19
`YoungAdult` for *age* between 20 – 29
`Adult` for *age* between 30 – 44
`Mature` for *age* between 45 – 64
`Senior` for *age* between 65 – 79
`Elderly` for *age* 80 and over

Feature Creation: Splitting

Feature Splitting

Example: Name split to FirstName, LastName

Example: Date 2019-06-21 split to Year, Month, Day

Python featuretools

name	type	description
num_true	aggregation	Finds the number of 'True' values in a boolean.
percent_true	aggregation	Finds the percent of 'True' values in a boolean feature.
time_since_last	aggregation	Time since last related instance.
num_unique	aggregation	Returns the number of unique categorical variables.
avg_time_between	aggregation	Computes the average time between consecutive events.
all	aggregation	Test if all values are 'True'.
min	aggregation	Finds the minimum non-null value of a numeric feature.
mean	aggregation	Computes the average value of a numeric feature.
seconds	transform	Transform a Timedelta feature into the number of seconds.
second	transform	Transform a Datetime feature into the second.
and	transform	For two boolean values, determine if both values are 'True'.
month	transform	Transform a Datetime feature into the month.
cum_sum	transform	Calculates the sum of previous values of an instance for each value in a time-dependent entity.
percentile	transform	For each value of the base feature, determines the percentile in relation
time_since_previous	transform	Compute the time since the previous instance.
cum_min	transform	Calculates the min of previous values of an instance for each value in a time-dependent entity.

Feature Creation: Other

Clusters

PCA Components

Flags

flag variables indicating missing values
flag variables indicating outliers
etc

Feature Contribution

Correlation Example

r^2 measures how much of variation is explained by linear regression

Contribution to Model

When building a model from your dataset,
does the technique allow you
to know the contribution of each feature?

Compare with PCA

PCA finds principal orthogonal components
components are ranked by contribution
components are defined as combinations of features