Feature Engineering using IBM Watson Studio

Tsun Chow

Agenda

- What is feature engineering?
- Common Engineering techniques
- Demo with Watson Studio with the Titanic Dataset

Feature Engineering

- Is the process of figuring out the best way of representing the data to machine learning algorithms to improve prediction

Feature Engineering

‘**Feature engineering** is the process of using domain knowledge of the data to create features that make machine learning algorithms work.”

--- *Wikipedia*

‘Coming up with features is difficult, time-consuming, requires expert knowledge. "Applied machine learning" is basically feature engineering.’

— *Andrew Ng*

Common Engineering techniques

- One-Hot coding
- Binning
- Normalization
- Standardization
- Dealing with missing values
- Data imputation techniques

One-Hot coding

- Some algorithms only works with numerical values.
- Features with Categorical values must be converted to numerical values.
- One-hot coding
  - uses a dummy binary variable to represent each categorical value
  - Assign 1 if the color is present (hot) and 0 otherwise

Example of One-Hot coding

- Example: color: red, yellow and green
- Introduce 3 binary dummy variables:
- [red, yellow, green]
- If for color = red.it is now represented as [1, 0, 0]
- Color values should not be coded as ordered numbers such as 1, 2, 3
- This ordinal version will not increase the dimensionality but may confuse some algorithms.

Binning

- Convert feature with continuous values into multiple binary features by bucketing based on value range.
- For example, the birth year of a person. The specific value of the year may not matter but what generation that it falls in may matter
- Based on frequency or other grouping that makes semantic sense: child, teens, adult, seniors
- It is a form of “regularization” or “smoothing”. It reduces the chance of “overfitting”

Binning of Birth Year by Generation

- Create 5 binary dummy features in this order
  - Gen Z, iGen, or Centennials: Born 1996 – 2019.
  - Millennials or Gen Y: Born 1977 – 1995.
  - Generation X: Born 1965 – 1976.
  - Baby Boomers: Born 1946 – 1964.
  - Traditionalists or Silent Generation: Born 1945 and before
- For someone born in 2001, the birth age feature is now coded as [1, 0, 0, 0, 0]

Feature Rescaling

- Value ranges for features could vary significantly
- It is not necessary but it would speed up the optimization of the algorithms
- Many algorithms may be able to handle the vast differences in feature ranges
- 2 common methods:
  - Normalization
  - Standardization

Nominalization & Standardization

- Nominalization:
  - Converting the actual values into a standard range of values typically [-1, 1] or [0, 1]
  - New value = (original value – mean)/(max-min)
- Standardization (Z score):
  - Converting the actual values so that the mean = 0 and standard deviation =1 from the mean
  - New value = (original value – mean) / standard deviation

Rules of Thumb for when to use which scaling*

- Use standardization for the following cases:
  - Unsupervised learning algorithms
  - If the feature value distribution closes to a normal distribution
  - Feature values have extremely high or low values (outliers)
- Use normalization for all other cases.

* see reference 1

Handling Missing Values

- Dropping the observations that have missing values if you have enough data left
- Use learning algorithms that can handle missing values
- Use a data imputation technique

Data Imputation Techniques

- Replace the missing value with the mean of the values of this feature
- Replace the missing value with the value that occurs with the highest frequency of this feature
- Replace the missing value with a value that is way outside the range.
  - If the range is [0, 1] set it to -1 or 2
- Replace the missing value with a value that is in the middle of the range.
  - If the range is [-1. 1] set it to 0
- Replacing the missing values with a random number with mean = average and and sd = sd of the non-missing values

Data Imputation Techniques

- Use features with no-missing values as the data to predict the missing value of this feature by using regression
- If the dimensionality of the dataset is low, add a new dummy variable D to the observation:
  - D = 0 if the feature value is missing , 1 otherwise
  - The missing value itself can be set to 0 or any number.

Which Data Imputation Technique to use?

- You cannot tell in advance
- The only is to build models with several different techniques to see which model works the best

Demo

- Data Refinery in Watson Studio
- Titanic Dataset

Watson Studio Data Refinery

- Built on top of the plyr R package
- Tool for analyzing and transforming data
- 100 plus built-in operations w/o programming
- Profile and visualize data
- Allow sampling for large data sets
- Use data flow to record, iterate and run steps in data refining

Titanic Dataset

- Has a popular movie with the same name
- You all have some domain knowledge of ships and passengers relative to survival in the event of an accident.
- One of the most studied dataset in machine learning with > 13,000 users have used it to build prediction models in Kaggle

The RMS Titanic sank in the early morning of 15 April 1912 in the North Atlantic Oce
Titanic had an estimated 2,224 people on board. More than 1,500 people died.

Titanic the Movie

https://youtu.be/2e-eXJ6HgkQ

The Titanic (Training) Data set

- About 891 observations
- 12 features
- Goal: to prepare the data for training a model for survival prediction (supervised learning)
- Analysis requires domain knowledge: meaning of the feature and knowledge of the incident
- Transform features using Data Refinery

Titanic Dataset

- PassengerID: id number of the passenger
- survival : Survival (0 = No; 1 = Yes)
- Pclass: Passenger Class (1 = 1st; 2 = 2nd; 3 = 3rd)
- Name: Name of the passenger with title
- Sex: Gender of the passenger: male or female
- Age: in years
- Sibsp: Number of Siblings/Spouses Aboard
- Parch: Number of Parents/Children Aboard ticket
- Ticket Number: ticket id, not unique (a group may share the same ticket id)
- Fare: Passenger Fare (British pound) per ticket (not per person)
- Cabin: Cabin Number
- Embarked: Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

Demo

Refinery Features and Their Use

- Profiling:
  - To detect Duplicates
  - To detect Missing or empty values
- Visualization: histograms:
  - Survived split by Sex
  - Pclass split by SurvivedCategory
  - Age split by SurvivedCategory

Refinery Features and Their Use

- Convert a column type:
  - Survived from string into category, other columns from string to decimal or integer as needed
- Conditional Replace for One-Hot coding:
  - Embarked into binary Embarked_S, Embarked_C, Embarked_Q
- Calculations:
  - Add Two Columns into One: add Parch to Sibl and store the sum in a new column, Relative
  - Add 1 to Relative to create Familysize
  - Divide Familysize into Fare to create Fareperperson???

Refinery Features and Their Use

- Replacing missing values
  - Fill missing value in Age

- Conditional Replace with “is empty” for replacing empty values:
  - Filling empty value for Embarked

Data Extraction
(Conditional Replace with “contains” as the logical condition)

- Extract the Deck (First letter) from Cabin
  - "A": 1, "B": 2, "C": 3, "D": 4, "E": 5, "F": 6, "G": 7, "U": 8} Missing values 0
- Extract title from Name:
  - Code “Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Rare": 5}
  - Replace 'Mlle‘ for 'Miss‘,'Ms', for'Miss‘, 'Mme‘ for 'Mrs'

Titanic Feature Engineering

Feature	Old Type	Drop?	New Type	Missing Values	Feature Transformation
PassID	String	yes			Eliminate duplicate
Survived	String		Integer		
pClass	String		Integer		
Name	String	yes			Extract Title, integer
Sex	String	yes			Binary coding, Sex_Male
Age	String	yes		Random around a mean	Binning into AgeGroup,integer
Sibsp	String	yes			Combine with Parch
Parch	String	yes			To create Relative, integer
Ticket Number	String	yes			
Fare	String	yes			Create Fareperperson??? decimal
Cabin	String	yes		0	Extract Deck. integer
Embark	String	yes		S	One hot-coding, integer

Feature Engineering Takeaways

- depends on domain knowledge to choose the best technique for data transformation
- Requires experimentations
- Requires iteration (data flow very useful)
- Data Visualization helps
- A point-click interface makes the process easier and more effective (with limitations)

References

- “The Hundred –Page Machine Learning Book” by Andriy Burkov
- Data preparation for Titanic tutorial:

https://developer.ibm.com/tutorials/data-preparation-with-ibm-data-refinery/

- Titanic dataset analysis

https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8

References

- Titanic Top 4% with ensemble modeling

https://www.kaggle.com/yassineghouzam/titanic-top-4-with-ensemble-modeling

- This will help you score 95 percentile in the Kaggle Titanic ML competition

https://medium.com/@praveen.orvakanti/this-will-help-you-score-95-percentile-in-the-kaggle-titanic-ml-competition-aa2b3fd1b79b