Features Engineering & Selection

Zephania Reuben & Davis David

April 10, 2021

Contents

1. Feature Engineering

2. Missing Data

3. Continuous Features

4. Categorical Features

5. Feature Selection

Feature Engineering

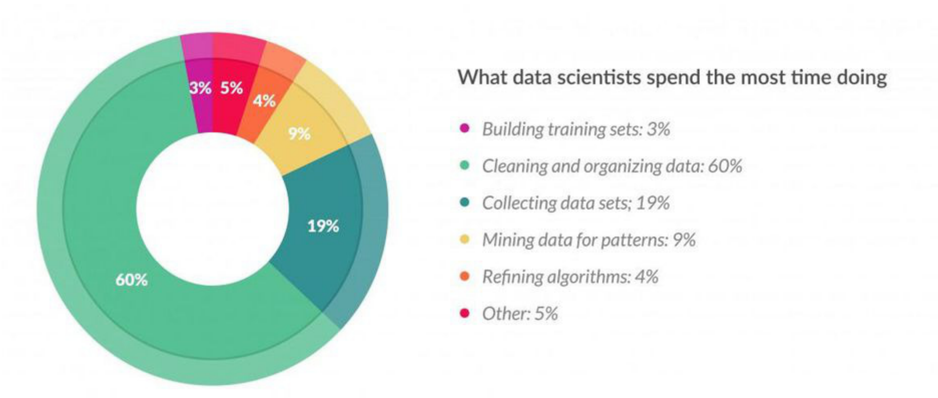
Feature engineering refers to a process of selecting and transforming variables/features when creating a predictive model using machine learning.

Feature engineering has two goals:

1. Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
2. Improving the performance of machine learning models.

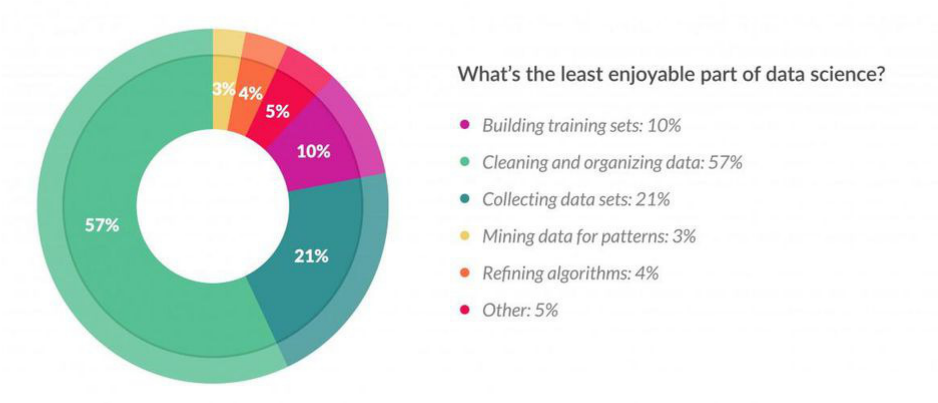
Feature Engineering

Data scientists spend 60% of their time on cleaning and organizing data.



Feature Engineering

57% of data scientists regard cleaning and organizing data as the least enjoyable part of their work.



Feature Engineering

"At the end of the day, some machine learning projects succeed and some fail.What makes the difference? Easily the most important factor is the features used."

- Prof. Pedro Domingos from University of Washington

Read his paper here: A few useful things to know about machine learning

Missing Data

- Handling missing data is important as many machine learning algorithms do not support data with missing values.
- Having missing values in the dataset can cause errors and poor performance with some machine learning algorithms.

Missing Data

- Common missing values.
  - N/A
  - null
  - Empty
  - ?
  - none
  - empty
  - -
  - NaN

How to handle Missing Values

1. Variable Deletion

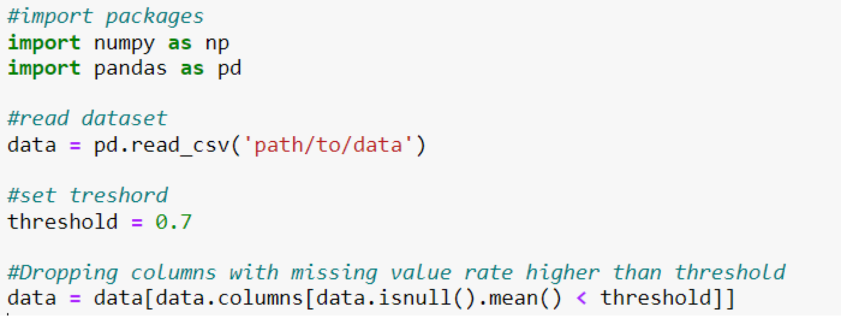
   - Variable deletion involves dropping variables(columns) with missing values on an case by case basis.

   - This method makes sense when lot of missing values in a variable and if the variable is of relatively less importance.

   - The only case that it may worth deleting a variable is when its missing values are more than 60% of the observations.

How to handle Missing Values

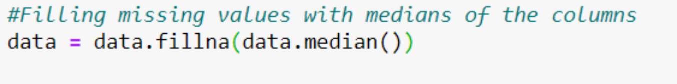
1. Variable Deletion



```
#import packages
import numpy as np
import pandas as pd

#read dataset
data = pd.read_csv('path/to/data')

#set treshord
threshold = 0.7

#Dropping columns with missing value rate higher than threshold
data = data[data.columns[data.isnull().mean() < threshold]]
```

How to handle Missing Values

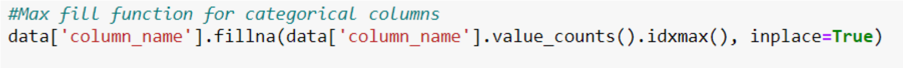
- Mean or Median Imputation
  - A common technique is to use the mean or median of the non-missing observations.
  - This strategy can be applied on a feature which has numeric data.



```
#Filling missing values with medians of the columns
data = data.fillna(data.median())
```

How to handle Missing Values

- Most Common Value
  - Replacing the missing values with the maximum occurred value in a column/feature.This is a good option for handling categorical columns/features.



```
#Max fill function for categorical columns
data['column_name'].fillna(data['column_name'].value_counts().idxmax(), inplace=True)
```

Continuous Features

- Continuous features in the dataset have different range of values.
- If you train your model with different range of value the model will not perform well. Example continuous features: age, salary , prices, heights
- Common methods.
    - Min-Max Normalization
    - Standardization

Continuous Features

1. Min-Max Normalization

   For each value in a feature, Min-Max normalization subtracts the minimum value in the feature and then divides by the range. The range is the difference between the original maximum and original minimum.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

It scale all values in a fixed range between 0 and 1.

Continuous Features

1. Standardization

   The Standardization ensures that for each feature have the mean is 0 and the variance is 1, bringing all features to the same magnitude.

   If the standard deviation of features is different, their range also would differ from each other.

   $$z = \frac{x - \mu}{\delta} \tag{2}$$

   $x =$ observation, $\mu =$ mean , $\delta =$ standard deviation.

Categorical Features

- Categorical features represents types of data which may be divided into groups. Example: genders, educational levels.
- Any non-numerical values need to be converted to integers or floats in order to be utilized in most machine learning libraries
- Common methods.
  - One-Hot-Encoding(Dummy Variables)
  - Label Encoding

Categorical Features

- One-hot-encoding

  By far the most common way to represent categorical variables is using the one-hot encoding or one-out-of-N encoding, also known as dummy variables.

  The idea behind dummy variables is to replace a categorical variable with one or more new features that can have the values 0 and 1.

Categorical Features

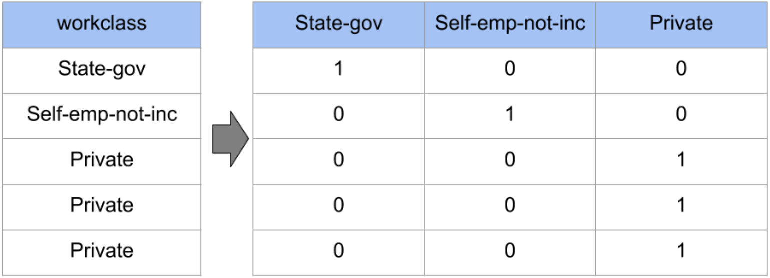
- One-hot-encoding

  By far the most common way to represent categorical variables is using the one-hot encoding or one-out-of-N encoding, also known as dummy variables.

  The idea behind dummy variables is to replace a categorical variable with one or more new features that can have the values 0 and 1.

Categorical Features

- One-hot-encoding

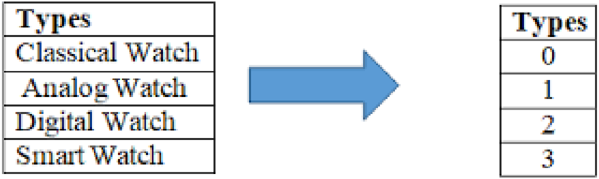


workclass
State-gov
Self-emp-not-inc
Private
Private
Private

State-gov	Self-emp-not-inc	Private
1	0	0
0	1	0
0	0	1
0	0	1
0	0	1

Categorical Features

- Label Encoding
  - Label encoding is simply converting each categorical value in a column to a number.



Types
Classical Watch
Analog Watch
Digital Watch
Smart Watch

Types
0
1
2
3

NB: It is recommended to use label encoding to a Binary variable

Feature Selection

- Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in.
- Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features.

Feature Selection

Top reasons to use feature selection are:

- It enables the machine learning algorithm to train faster.
- It reduces the complexity of a model and makes it easier to interpret.
- It improves the accuracy of a model if the right subset is chosen.
- It reduces overfitting.

Feature Selection

“I prepared a model by selecting all the features and I got an accuracy of around 65% which is not pretty good for a predictive model and after doing some feature selection and feature engineering without doing any logical changes in my model code my accuracy jumped to 81% which is quite impressive”

- By Raheel Shaikh

Feature Selection

- Univariate Selection

Statistical tests can be used to select those independent features that have the strongest relationship with the target feature in your dataset. E.g. Chi squared test.

The scikit-learn library provides the SelectKBest class that can be used with a
suite of different statistical tests to select a specific number of features.

Article: A Gentle Introduction to the Chi-Squared Test for Machine Learning

Feature Selection

- Feature Importance

Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your target feature.

Feature importance is an inbuilt class that comes with Tree Based Classifiers Example: Random Forest Classifiers and Extra Tree Classifiers

Feature Selection

- Correlation Matrix with Heatmap

  Correlation show how the features are related to each other or the target feature.

  Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable)