PreProcessing For ML

Cleaning / Preprocessing data for ML

@Shay Palachy
@dmarcous

Preso : https://goo.gl/q6a376
Colab : https://goo.gl/XNmkBW

Colab

Make sure to run everything by order!

- If you don’t have a number on the left it didn’t run

Setup

Imports

[1]
```
# On sucess - you get no output
import pandas as pd
import numpy as np
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
from sklearn import metrics
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

seed = 666
```

Feature Engineering

Feature Engineering

Giving the model features that are easy to learn from

- Required with models that can't model complex feature combinations
- Example :
  - Existing : x1 = house width, x2 = house length, x3 = house height
  - Engineered : x4 = house volume (x1*x2*x3)

Feature Engineering - Code

Scikit Learn

Transformers

Fit & Transform

- Learn on train, test on new data
    - E.g. mean imputation
- Same interface over all models / preprocessing transformers
- Easy to extend
- Easy to pipeline

Data Transformations

Giving the model data that is easy to learn from

Imputation

Completing Missing Values

- Drop missing values?
- Fill values
- Statistics based - average/ median / frequent
- Model based - predict missing value based on similar records

One Hot Encoding

The Standard Approach for Categorical Data

- creates new (binary) columns
- Beware of large number of categorical values (See Feature Hashing)

Color
Red
Red
Yellow
Green
Yellow

Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

Scaling and Normalization

- Scaling - Changing range
  - A change of 1 in feature A = change of 1 in feature B
  - For models that measure distance over features (KNN / SVM)
  - StandardScaler = x-mean/std (robust to outliers)
- Normalization - Changing shape
  - Make data follow a normal distributions
  - For models that assumes data is normally distributed (Linear Regression)
  - Can be useful for sparse datasets with attributes of varying scales (especially w/ algorithms weighing input values [NNs] and algorithms that use distance measures [KNN])
  - Can help restrain the effects of outliers

Scaling and Normalization

```
>>> from sklearn.preprocessing import Normalizer
>>> X = [[4, 1, 2, 2],
...      [1, 3, 9, 3],
...      [5, 7, 5, 1]]
>>> transformer = Normalizer().fit(X) # fit does nothing.
>>> transformer
Normalizer(copy=True, norm='l2')
>>> transformer.transform(X)
array([[0.8, 0.2, 0.4, 0.4],
       [0.1, 0.3, 0.9, 0.3],
       [0.5, 0.7, 0.5, 0.1]])
```

Anomaly Detection

- Remove / Tag anomalies
  - They don't reflect "real data"
  - Might cause us to learn very rare patterns and stray from the common patterns
- Methods
  - Range based - e.g. top 2% of most expensive houses
  - Model based - larger error , large distance from all other samples etc.

Feature Selection

Feature Selection

Selecting a subset of features to gain best performance

- Some features help prediction
- Some features are just “noise”
- If 2 features correlate - they might be “over represented” in the model
    - Daily drives & weekly drives
- Some concept of feature “importance” is needed

Each subset forms a new hypothesis : test model with features {f1,...,fm}

Feature Selection Methods

- Keep features with high variance
- Keep features correlated to the target variable
- Drop 1 of a pair of correlated features
- Keep important features - e.g. via “Information gain”
- **Keep features “important” in a previous model (model.feature_importances_)**
- Brute force - check all subsets (usually unreasonable - 2^n)

Dimensionality Reduction

Dimensionality

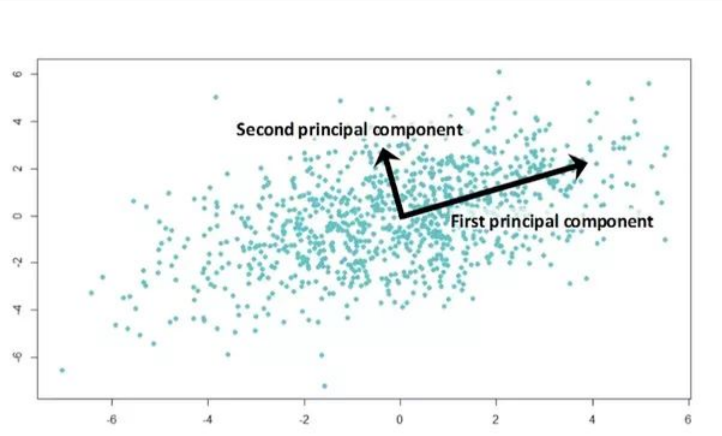
Feature = Dimensions (in “feature space”)

- Takes many compute resources
- Patterns are complex, thus hard to learn

Can we reduce dimensions while not losing any knowledge?

Principal Component Analysis

- Each component is a linear combination of original features
- Maximizes variance in original dataset (=retains knowledge)

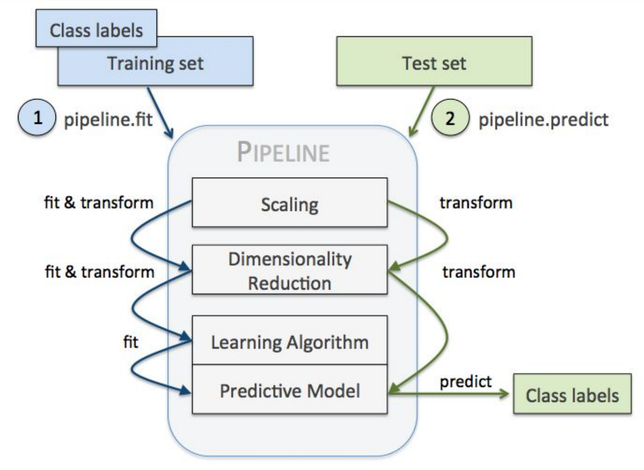


ML Pipeline

Scikit Pipeline

Sequentially apply a list of transforms and a final estimator

- Cleaner code
- Easier to productionize



Pipeline - All in One

Code