FEATURE SELECTION

Motivation

- Performance could degrade when including input variables that are not relevant to the target variable.
- Overfitting for tasks with a smaller # of samples
- A large number of variables can be computationally expensive

Typical techniques

- Remove features with low variance
- Remove features with low correlation based on statistical tests
- Sequential feature selection
  - Forward: iteratively add the best new features
  - Backward: iteratively remove the least useful feature
- https://scikit-learn.org/stable/modules/feature_selection.html

Feature Engineering

- Very different from feature selection
- Example: predict housing price
- Input: square footage, transaction date, built date, and price

Feature Engineering

- Very different from feature selection
- Example: predict time-to-sell of a house
- Input (features and label): square footage, lot size, transaction date, built date, and price
- Engineered features could include
  - Cost per sq. ft
  - House age
  - Zip code
  - School rating
- Data preprocessing (e.g., normalization, missing data) sometimes are also considered as feature engineering

Typical process

- Brainstorm features
- Deciding what features to create
- Creating features
- Testing the impact of the identified features on the task
- Improving your features if needed
- Repeat

Features

- Feature selection
- Feature engineering
- PCA
- Differences

ML PRACTICES

Be Cautious

- AI/ML is not a cure-all
- "All models are wrong, some are useful."
  –George Box
- Understand your models, know the assumptions and limitations of the models
- Is AI a hype or a GE?

Typical steps to apply ML

- Data preprocessing
- Trying different ML algorithms
  - Training set, validation set, test set
- Diagnostics
  - More training samples
  - Increase/decrease feature set
  - Increase/decrease regularization
- Loop back

A ML Project

- Why ML is a suitable approach
  - Do not use ML for the purpose of using ML
  - Evaluate existing approaches and room for improvement
- Problem abstraction and formulation
  - Set appropriate goals
  - Model complexity, data availability, evaluation
  - Domain knowledge critical
- Data collection and data cleaning
  - What, where, and how
- ML algorithms
  - This is often the "easy" part

Characteristics of Good Problems

- Existing solutions not satisfactory
  - Automate the process
  - Improve performance
- Data availability: suitable data available or obtainable
- Data quality and quantity
- Can evaluate proposed approaches
- Large complex problem beyond white-box modeling
- Understanding complex venue and large data