Text Mining

Instructor: Jing Wang
Department of ISOM
Spring 2023

HKUST BUSINESS SCHOOL
香港科大商學院

Text Mining

- It’s estimated that 40% data mining tasks involve analysis of textual document.
- Companies and firms collect large amount of textual data from various sources
  - Social media platforms
  - Customer reviews
  - Financial report
  - …

2013 AP Twitter Hack

- The Associated Press is a major news agency that distributes news stories to other news agencies.
- In April 2013 someone tweeted the above message from the main AP verified Twitter account.
- The S&P500 stock index fell 1% in seconds, but the White House rapidly clarified.

Text Mining Applications

- Spam filtering: spam vs non-spam
- Article categorization: politics, sports, entertainments, travel, others
  - “The White House announced plans to introduce tariffs on US$60 billion worth of Chinese imports…”
- Sentiment analysis: positive, neutral, negative
  - “With just 3 months the battery is NOT working now. Very bad experience”

Dealing With Text

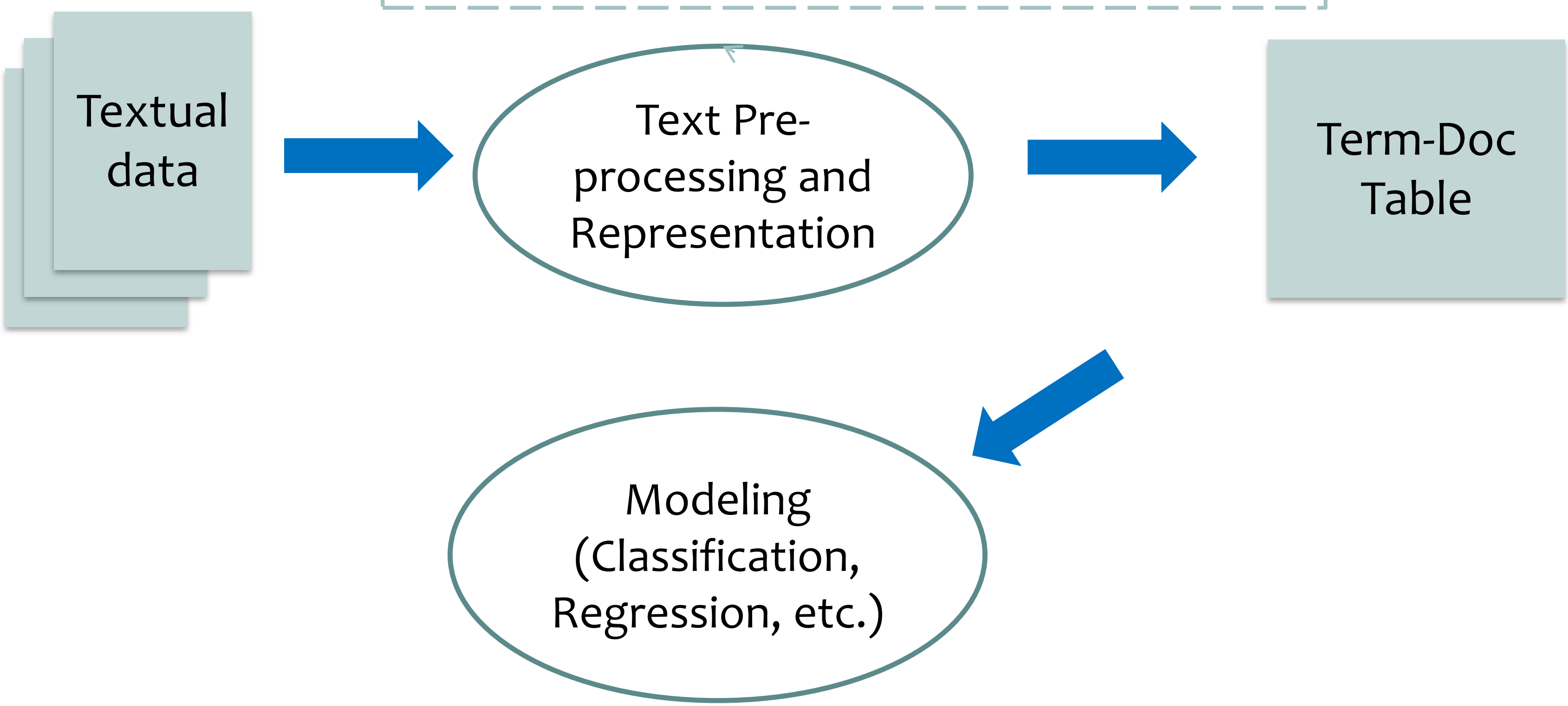
- Until now, our data has typically been structured
  - Numeric
  - Categorical

- Textual data
  - Loosely structured
  - Poor spelling, non-traditional grammar
- Vast amount of text
  - Humans can't keep up with Internet-scale volumes of data, e.g., ~500 million tweets per day!

But people care about textual data, how do we handle it?

Text Mining Workflow

- Stemming
- Removing stop words, punctuations
- Bag of words, n-grams, …
- 0/1, tf, tf-idf, …
- etc…

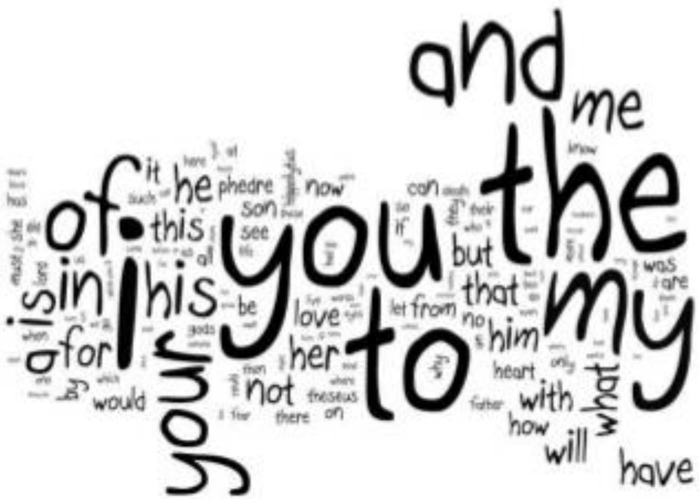


Pre-Processing of Text (I)

- Many words are frequently used but are only meaningful in a sentence - “**stop words**”
  - Examples: the, is, at, which, and, of, on…
  - Unlikely to improve prediction quality
  - Remove to reduce size of data

Can you think of a simple way to remove the stop words?

Build stop words list, whenever the words in the list occur remove it



Pre-Processing of Text (II)

- Do we need to draw a distinction between the following words? e.g., argue argued argues arguing
  - Could all be represented by a common stem, argu
- Stemming: the process of reducing inflected (or sometimes derived) words to their word stem.
  - ran, running, runs => run
  - loved, lovely, loving => love

Generate common rules - useful but not solve all problem

Can you think of a simple way for computer programs to do stemming?

Pre-Processing of Text (III)

- Remove punctuations
  - Punctuation: e.g., @, #, !
  - Remove everything that is not a, b, …, z
- Lowercase all words
  - HKUST, Hkust => hkust

Text Representation: “Bag of Words”

- Each document is one instance/example
- Treat every document as just a collection of individual words
  - Ignore grammar, word order, sentence structure, and punctuation
  - Every word in a document is a feature
- Straightforward representation; inexpensive to generate
- Tends to work well for many tasks

What will be the values of features in a given document?

Document Representation 1: 0/1

--- Each entry in the table represents a document.

--- Attribute describes **whether or not** a term appears in the document.

Data mining is the process of extracting patterns from data. As more data are gathered, with the amount of data doubling every three years, data mining is becoming an increasingly important tool to transform these data into information. It is commonly used in a wide range of profiling practices, such as marketing, surveillance, fraud detection and scientific discovery…

	Term				
	Data	Mining	Tool	XXX	…
Document 1	1	1	1	0	
Document 2	1	0	0	0	
…	…	…	…	…	…

Term-Doc Table

Statistics is a branch of mathematics concerned with collecting and interpreting data. According to other definitions, it is a mathematical science pertaining to the collection, analysis, interpretation or explanation, and presentation of data.

Document Representation 2: Term Frequency

--- Each entry in the table represents a document.

--- Attribute describes the **frequency** of a term in the document.

--- *Usually needs normalized by length.*

Data mining is the process of extracting patterns from data. As more data are gathered, with the amount of data doubling every three years, data mining is becoming an increasingly important tool to transform these data into information. It is commonly used in a wide range of profiling practices, such as marketing, surveillance, fraud detection and scientific discovery…

	Term				
	Data	Mining	Tool	XXX	…
Document 1	6	2	1	0	
Document 2	2	0	0	0	
…	…	…	…	…	…

Term-Doc Table

Statistics is a branch of mathematics concerned with collecting and interpreting data. According to other definitions, it is a mathematical science pertaining to the collection, analysis, interpretation or explanation, and presentation of data.

Normalizing Term Frequency

- Documents of various lengths
  - Longer documents tend to be associated with higher term frequency
- Words of different frequencies
  - Words should not be too common
- A word is a good feature if it appears frequently in the document but is infrequent in the entire document set (corpus).
  - Solution: normalize the raw term frequencies in some way, e.g., TF-IDF

TF-IDF

- TF (term frequency): the higher, the more representative a term is for a given document

$$\mathrm{TF}(t,d) = \frac{\text{Number of occurences of term } t \text{ in document } d}{\text{Total number of words in document } d}$$

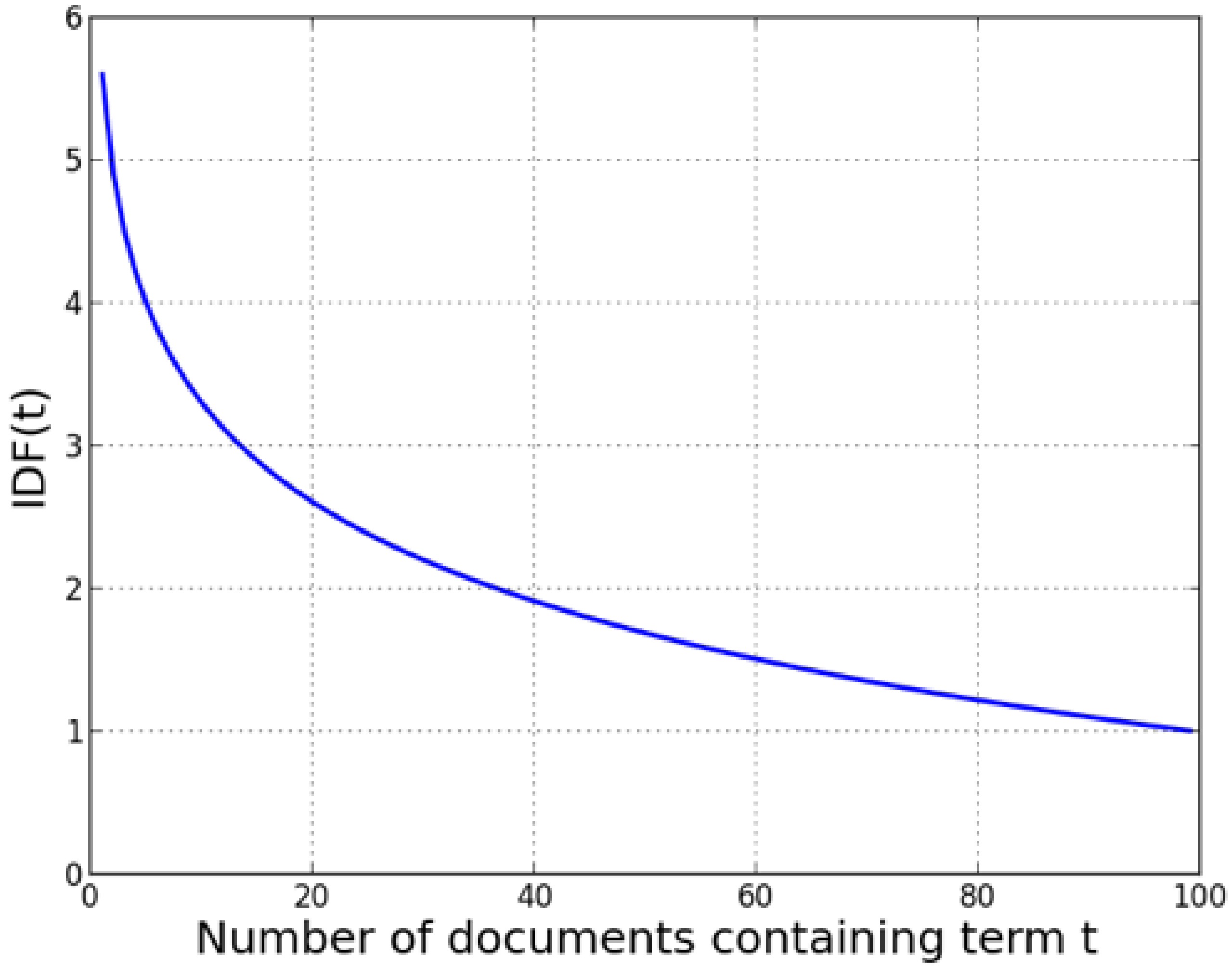
- IDF (inverse document frequency): the higher, the more effective the term is in differentiating documents from each other

$$\mathrm{IDF}(t) = 1 + \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing } t}\right)$$

- A simple measure that combines the two aspects:

$$\boxed{\mathrm{TF{-}IDF}(t,d) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t)}$$

IDF of a Term



0
1
2
3
4
5
6
IDF(t)
0
20
40
60
80
100
Number of documents containing term t

Exercise

- Suppose that we have term count tables of a corpus consisting of only two documents, as listed below.

Document 1

Term	Term Count
this	1
is	1
a	2
sample	1

Document 2

Term	Term Count
this	1
is	1
another	2
example	3

- What is the TF-IDF weights for "example" in Document 1 and Document 2, respectively?

0

3/7 * 1.69 =

Document Representation 3: TFIDF

--- Each entry in the table represents a document.

--- Attribute represents the **TF-IDF** of a term

Data mining is the process of extracting patterns from data. As more data are gathered, with the amount of data doubling every three years, data mining is becoming an increasingly important tool to transform these data into information. It is commonly used in a wide range of profiling practices, such as marketing, surveillance, fraud detection and scientific discovery…

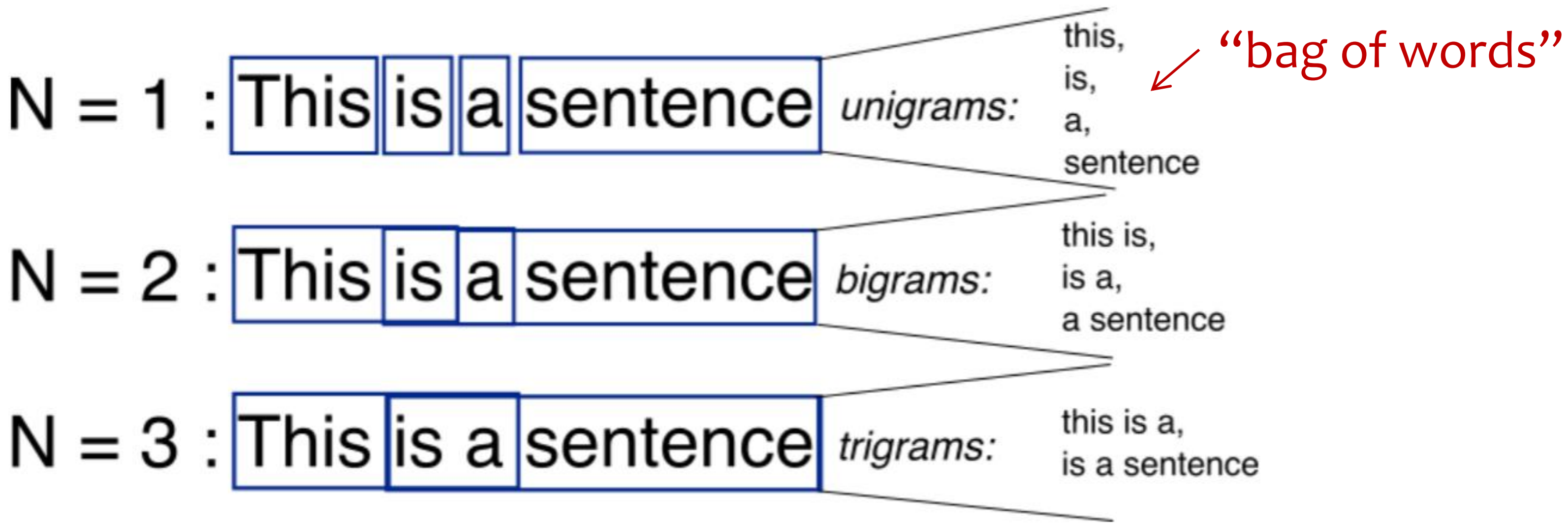
	Term				
	Data	Mining	Tool	XXX	…
Document 1	0.006	0.04	0.005	0	
Document 2	0.002	0	0	0	
…	…	…	…	…	…

Term-Doc Table

Statistics is a branch of mathematics concerned with collecting and interpreting data. According to other definitions, it is a mathematical science pertaining to the collection, analysis, interpretation or explanation, and presentation of data.

Text Representation: N-Grams

- N-grams: a contiguous sequence of n tokens from a given piece of text



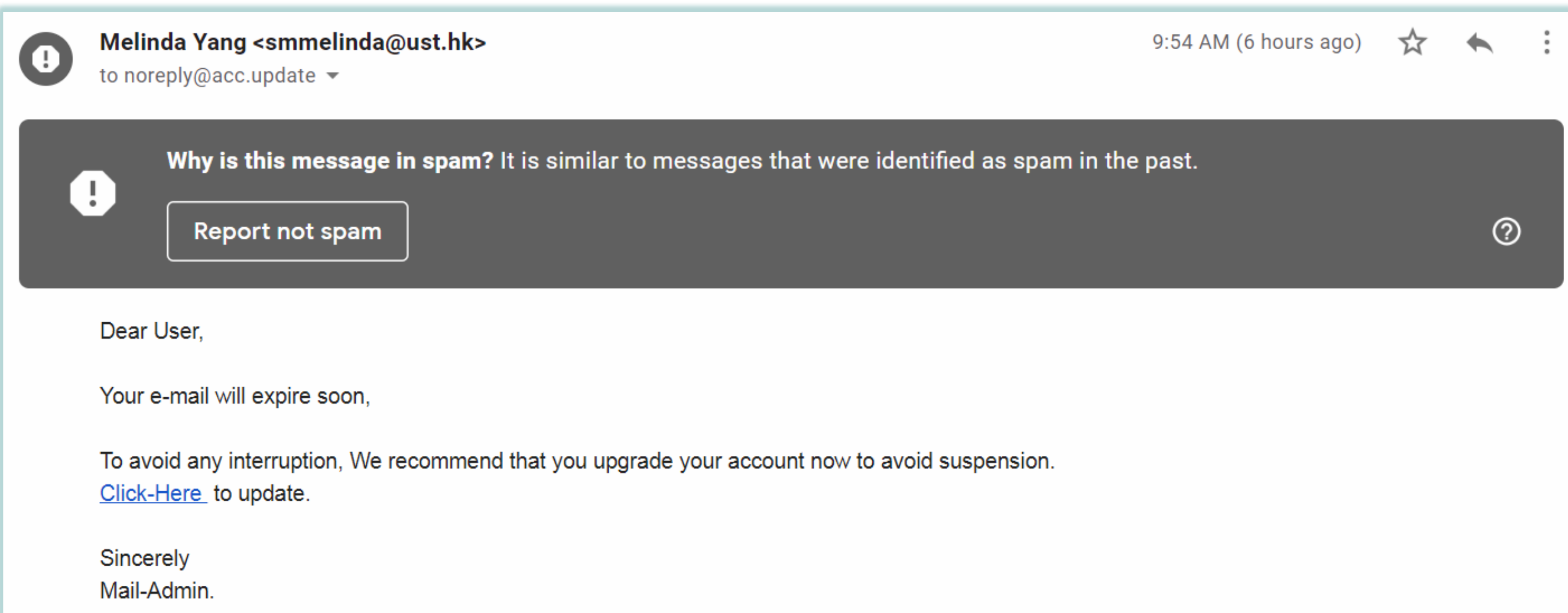
Naïve Bayes Summary: Strengths

- Strengths
  - Simple; efficient in both storage space and computation time
  - Performs well for classification in many real-world applications
  - Incremental learning (no need to reprocess all past training data when new data become available)

Naïve Bayes classifiers work quite well in many real-world situations, famously document classification and spam filtering.

A Text Mining Application: Spam Detection

- In 2014, there are estimated to be 4.1 billion email accounts worldwide, and about 196 billion emails are sent each day worldwide.
- Spam is one of the major threats posed to email users. In 2013, 69.6% of all email flows were spam.


Melinda Yang <smmelinda@ust.hk>
to noreply@acc.update
9:54 AM (6 hours ago)

Why is this message in spam? It is similar to messages that were identified as spam in the past.

Report not spam

Dear User,

Your e-mail will expire soon,

To avoid any interruption, We recommend that you upgrade your account now to avoid suspension.
Click-Here to update.

Sincerely
Mail-Admin.

Naïve Bayes Classifier

- Want $P(spam|words), P(ham|words)$
- Use Bayes Rule: $P(spam|words) = \frac{P(words|spam)P(spam)}{P(words)}$
- Assume conditional independence: probability of each word independent of others, given class label

$$P(words \mid spam) = P(word1 \mid spam) \times P(word2 \mid spam) \times \ldots \times P(wordn \mid spam)$$

Among 1000 spams, "lottery" appears in 90 of them, then P("lottery"|spam)=??? 90/1000 = 9%

Among 2000 ham emails, "lottery" appears in 20 of them, then P("lottery"|ham)= ??? 20/2000 = 10%

Represent Documents (0/1)

index	Document	Label
1	free iPhone	Spam
2	free to come by today	Ham
3	lottery win by today	Spam

Vocabulary size=8

free	iPhone	to	come	by	today	lottery	win	label
1	1	0	0	0	0	0	0	Spam
1	0	1	1	1	1	0	0	Ham
0	0	0	0	1	1	1	1	Spam

Term-Doc table

P(free|Spam)=1/2 P(free|Ham)=1/1

P(iphone|spam)=1/2 P(iphone|Ham)=0/1

P(lottery|Spam)=1/2 P(lottery|Ham)=0/1

P(Ham)=1/3
P(Spam)=2/3

Estimate Posterior Probability

$P(\text{Spam}|\text{“free iPhone lottery”})$

$=P(\text{free}|\text{spam})P(\text{iPhone}|\text{spam})P(\text{lottery}|\text{spam})P(\text{spam})/P(\text{“free iPhone lottery”})$

$=0.5 \times 0.5 \times 0.5 \times 0.667 \,/P(\text{“free iPhone lottery”})$

$P(\text{Ham}|\text{“free iPhone lottery”})$

$=P(\text{free}|\text{ham})P(\text{iPhone}|\text{ham})P(\text{lottery}|\text{ham})P(\text{ham})/P(\text{“free iPhone lottery”})$

$=1 \times 0 \times 0 \times .333 \,/P(\text{“free iPhone lottery”}) = 0$

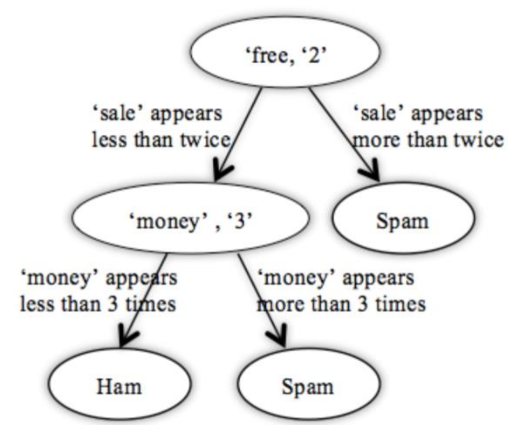
*For illustration purpose, need to apply Laplace smoothing to address zero-frequency problem.

More on Spam Detection

- In addition to word features, we can also include other features, such as the length of email, email address domain (.edu, .gov), etc.
- In practice, naïve Bayes spam detection has very high precision (~100%) and recall (~98.5%), with high AUC (~0.99)

Can we use other classification methods, such as decision tree or logistic regression?

Yes, as long as you have well structured dataset you can fit with different classification method



Lab: Spam Detection

- Files needed
  - TextMining_spam.ipynb (Python file)
  - spam.csv (dataset)

Feature Selection

Instructor: Jing Wang
Department of ISOM
Spring 2023

Feature Selection (I)

- What is feature selection?
  - A process of selecting a subset of relevant features for use in model construction.
- Why feature selection?
  - Simplified model is easier to interpret.
  - Reduce storage requirement and training time.
  - Reduce overfitting and achieve better generalization performance.

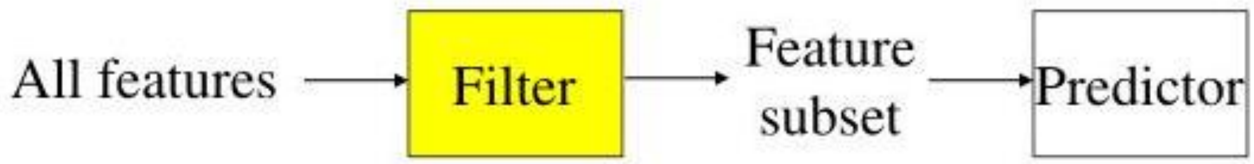
Feature Selection (II)

- What are "good" and "bad" features?
  - Good: in differentiating target variable.
  - Bad: redundant, irrelevant
- This is the most frequently asked question in real-world data analysis!!

Feature Selection Techniques

- Filter methods
  - Use a proxy measure to score features.
- Wrapper methods
  - Use the performance of a predictive model to score feature subsets.
- Embedded methods
  - A catch-all group of techniques which perform feature selection as part of the model construction process.

Filter Methods

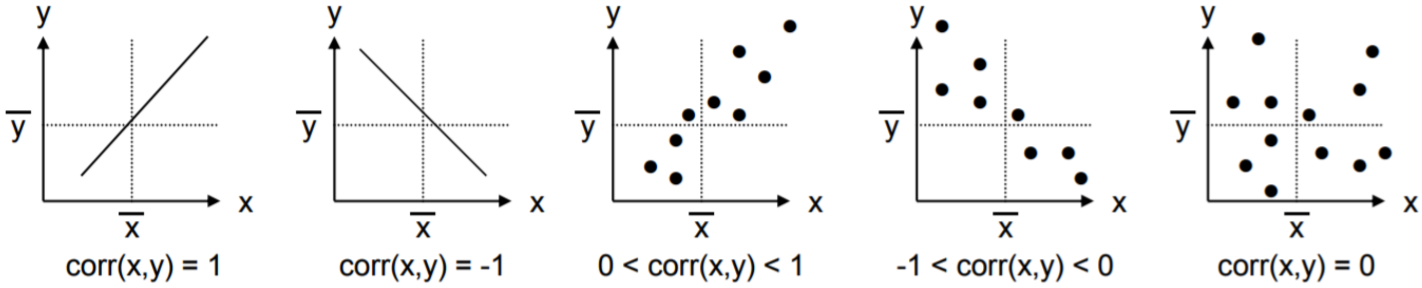


- Rank variables based on "relevance" and select features from the top of the list
- Assessment: statistical measures *regardless of the model*
  - Pearson correlation
  - Mutual information
  - Chi-square test
  - … (many others)

Filter Methods: Measures (I)

- Pearson Correlation: the linear correlation between the focal attribute and the target variable (We have seen this one in data understanding lecture!)

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

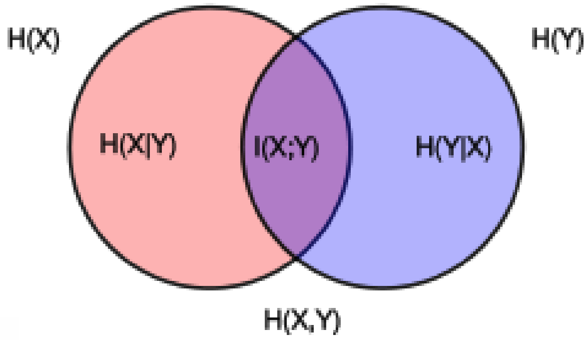


Support numeric data type!

Filter Methods: Measures (II) (Optional)

- Mutual Information: the mutual information between the focal attribute and the target variable (the contribution of the focal attribute towards reducing uncertainty about the value of target variable)
  - Categorical/discrete variable

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right)$$



  - Continuous variable

$$I(X;Y) = \int_Y \int_X p(x,y) \log\left(\frac{p(x,y)}{p(x)\,p(y)}\right) dx\,dy.$$

Support all data types!

Filter Methods: Measures (III) (Optional)

- Chi-Squared Test: a statistical method that measures how close expected values (if the focal attribute and the target variable are independent) are to actual results.

Number of actual observations

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

Number of expected observations if there is no relationship between the focal feature and target variable

Support categorical data type!

Filter Methods in Python

- For regression (mutual_info_regression, f_regression)
- For classification (mutual_info_classif, chi2, f_classif)

```
from sklearn.datasets import load_iris
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
iris = load_iris()
X, y = iris.data, iris.target
X.shape
```

```
(150, 4)
```

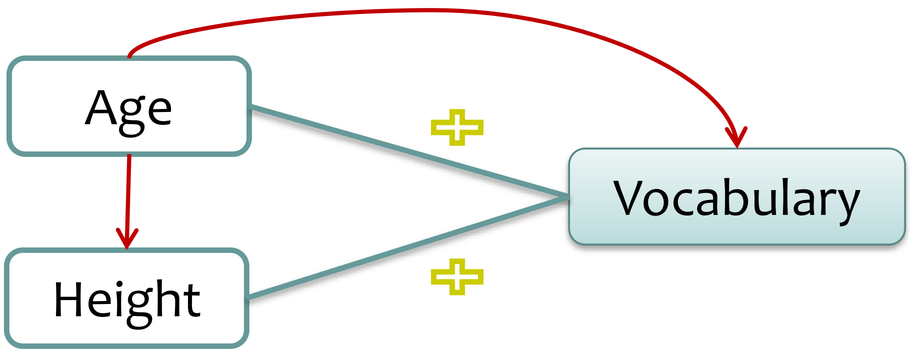
```
X_new = SelectKBest(chi2, k=2).fit_transform(X, y)
X_new.shape
```

```
(150, 2)
```

Use chi2 as the measure,
select the best 2 features

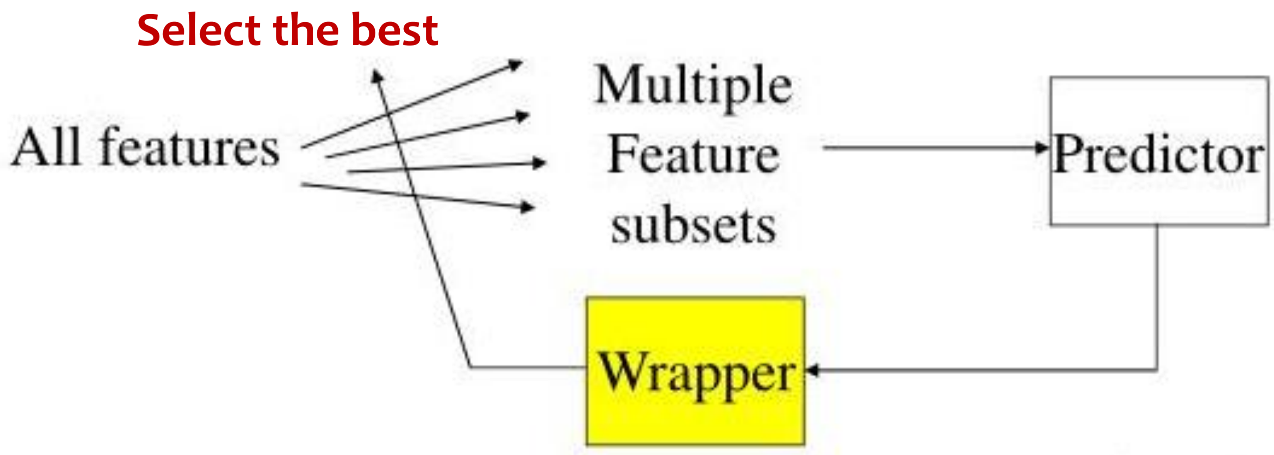
Filter Methods: Pros and Cons

- **Pros**: less computationally intensive; robust to overfitting
- **Cons**: tend to select redundant variables because they don't consider the relationships between variables; mainly used as a pre-process method
- Example of redundant variable:



Both age and height will be selected in predicting vocabulary of kids. But height will be redundant in the presence of age.

Wrapper Methods



- Try all possible feature subsets and measure feature subsets based on "usefulness"
- <u>Assessment</u>: use holdout validation or cross-validation
- **Pros**: model-oriented; usually have good performance for the model you choose
- **Cons**: very computationally expensive; prone to overfitting.

Number of Possible Feature Subsets (Optional)

If the total number of features is M, and we want to select half $(M/2)$ of the features. What is the total number of possible feature subsets?

$$\frac{M!}{\left(\frac{M}{2}\right)!\left(\frac{M}{2}\right)!}$$

--- For example, with 10 features, there are 252 possible 5-feature subsets; with 20 features, there are 184,756 possible 10-feature subsets.

Two Search Algorithms (Optional)

- Inclusion/Removal criteria uses cross-validation

Cross validation for evaluation

- Sequential forward selection
  - Start with no features
  - Greedily include the most helpful feature
- Sequential backward elimination
  - Start with all the features
  - Greedily remove the least helpful feature

Greatly reduce time in searching - efficient

What is the total number of possible feature subsets evaluated if we want to select half of the *M* features?

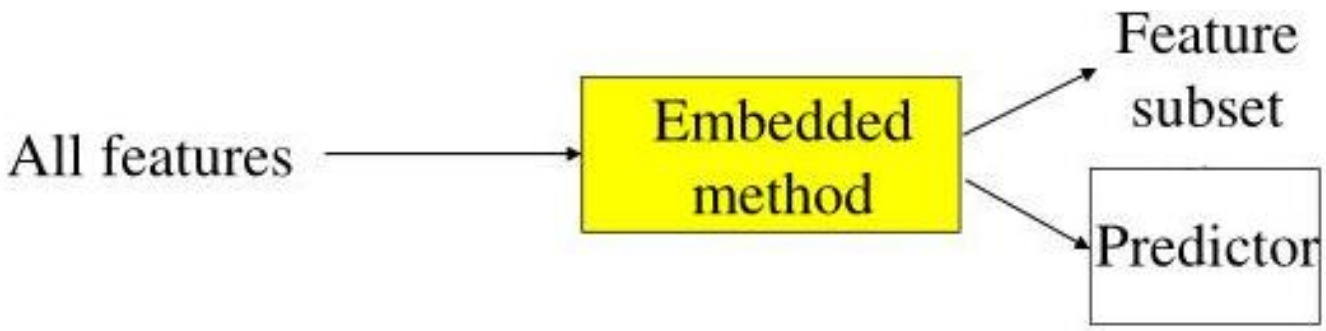
Wrapper Methods in Python

- Use the function *SequentialFeatureSelector* in python

```
>>> from sklearn.feature_selection import SequentialFeatureSelector
>>> from sklearn.neighbors import KNeighborsClassifier
>>> from sklearn.datasets import load_iris
>>> X, y = load_iris(return_X_y=True)
>>> knn = KNeighborsClassifier(n_neighbors=3)
>>> sfs = SequentialFeatureSelector(knn, n_features_to_select=3)
>>> sfs.fit(X, y)
SequentialFeatureSelector(estimator=KNeighborsClassifier(n_neighbors=3),
                          n_features_to_select=3)
>>> sfs.get_support()
array([ True, False,  True,  True])
>>> sfs.transform(X).shape
(150, 3)
```

Select the best 3 features

Embedded Methods



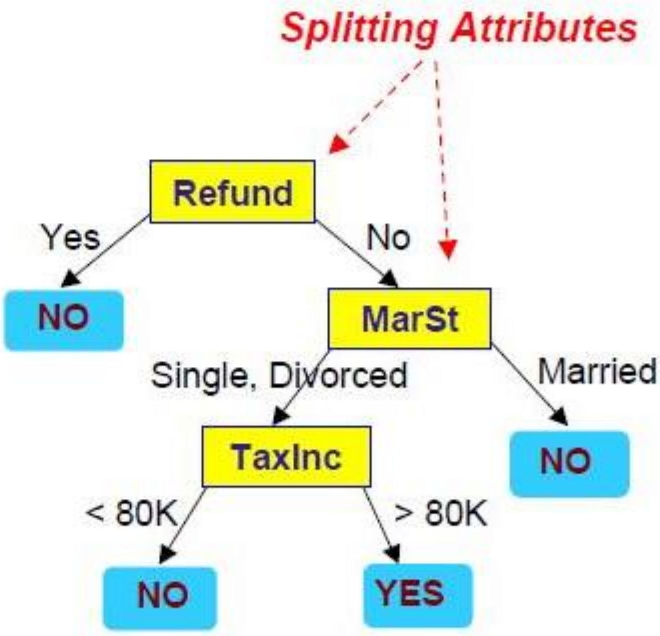
- Performs feature selection as part of the model building/ learning
- Computationally demanding (between filters and wrappers)

Embedded Methods: Decision Tree

- In final tree, only a subset of features are used, and the selected features are ordered based on how informative they are.

class

Tid	Refund	Marital Status	Taxable Income	Age	Cheat
1	Yes	Single	125K	18	No
2	No	Married	100K	30	No
3	No	Single	70K	28	No
4	Yes	Married	120K	19	No
5	No	Divorced	95K	18	Yes
6	No	Married	60K	20	No
7	Yes	Divorced	220K	25	No
8	No	Single	85K	30	Yes
9	No	Married	75K	20	No
10	No	Single	90K	18	Yes



Embedded Methods: L1-Regularization/Lasso

- In linear/logistic regression with L1-Regularization, features with non-zero coefficients are selected. And the importance of features are reflected by the magnitudes of corresponding coefficients (after feature normalization).

$$\min_{\boldsymbol{\theta}} \left[\underbrace{\sum_{i=1}^{n} \left(h_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) - y^{(i)}\right)^2}_{\text{model fit to data}} + \lambda \underbrace{\sum_{j=1}^{m} |\theta_j|}_{\text{regularization}} \right]$$

Compared to L2-regularization, L1-regularization is more aggressive about assigning a weight of 0 to features. It is useful in identifying features that can be removed.

Embedded Methods in Python

- Use the function *SelectFromModel* in python

```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier()
tree = tree.fit(X, y)
tree.feature_importances_
```

```
array([0.01333333, 0.        , 0.06405596, 0.92261071])
```

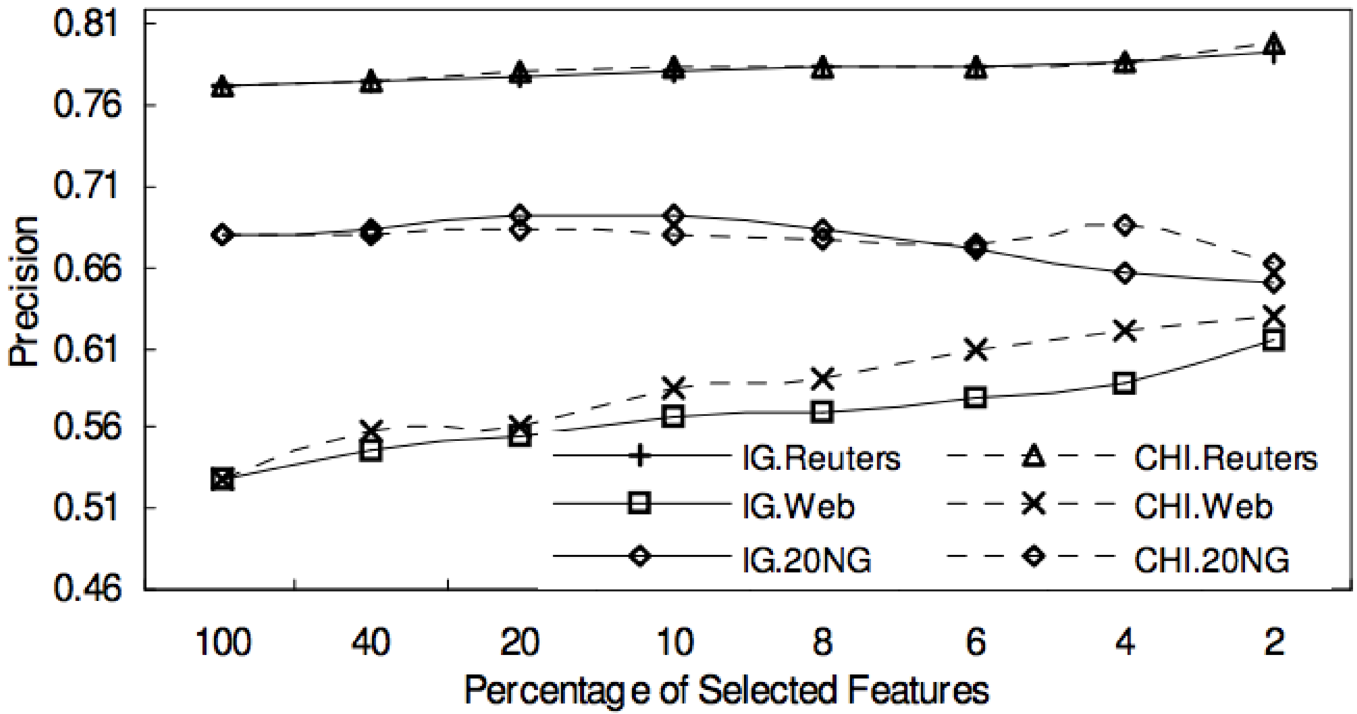
```
from sklearn.feature_selection import SelectFromModel
import numpy as np
model = SelectFromModel(tree, prefit=True,max_features=3,threshold=-np.inf)
X_new = model.transform(X)
X_new.shape
```

```
(150, 3)
```

Select the best 3 features

Example of Feature Selection in Text Classification

Data Sets	Classes Num.	Docs Num.	Terms Num
Reuters	80	10733	18484
20NG	20	18828	91652
Web	35	5035	56399



Lab: Feature Selection in Text Mining

- Files needed
  - FeatureSelection_spam.ipynb (python file)
  - spam.csv (dataset)

Hyperparameter Tuning

Instructor: Jing Wang
Department of ISOM
Spring 2023

HKUST BUSINESS SCHOOL
香港科大商學院

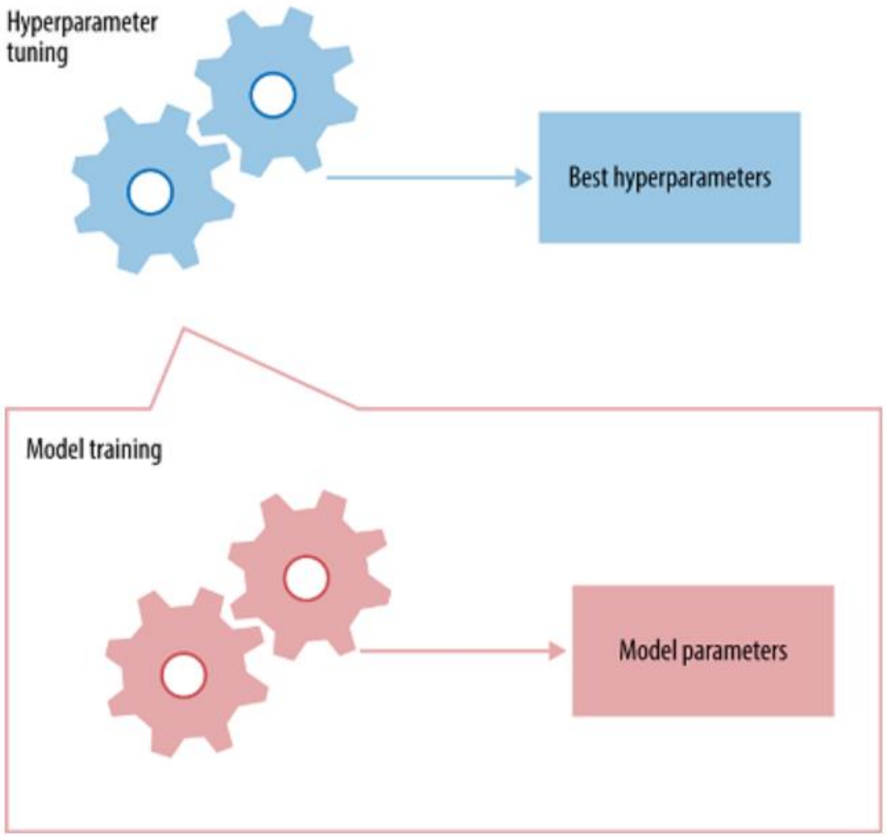
Parameters vs. Hyperparameters

- By training a model with existing data, we are able to fit/learn the **model parameters** (e.g., coefficients in linear regression, splitting attributes in decision tree, likelihoods in naïve Bayes classifier).

- **Hyperparameters** are usually fixed before the actual training process begins and cannot be directly learned from the regular training process (e.g., the depth of decision tree model, the regularization parameter in LASSO regression).

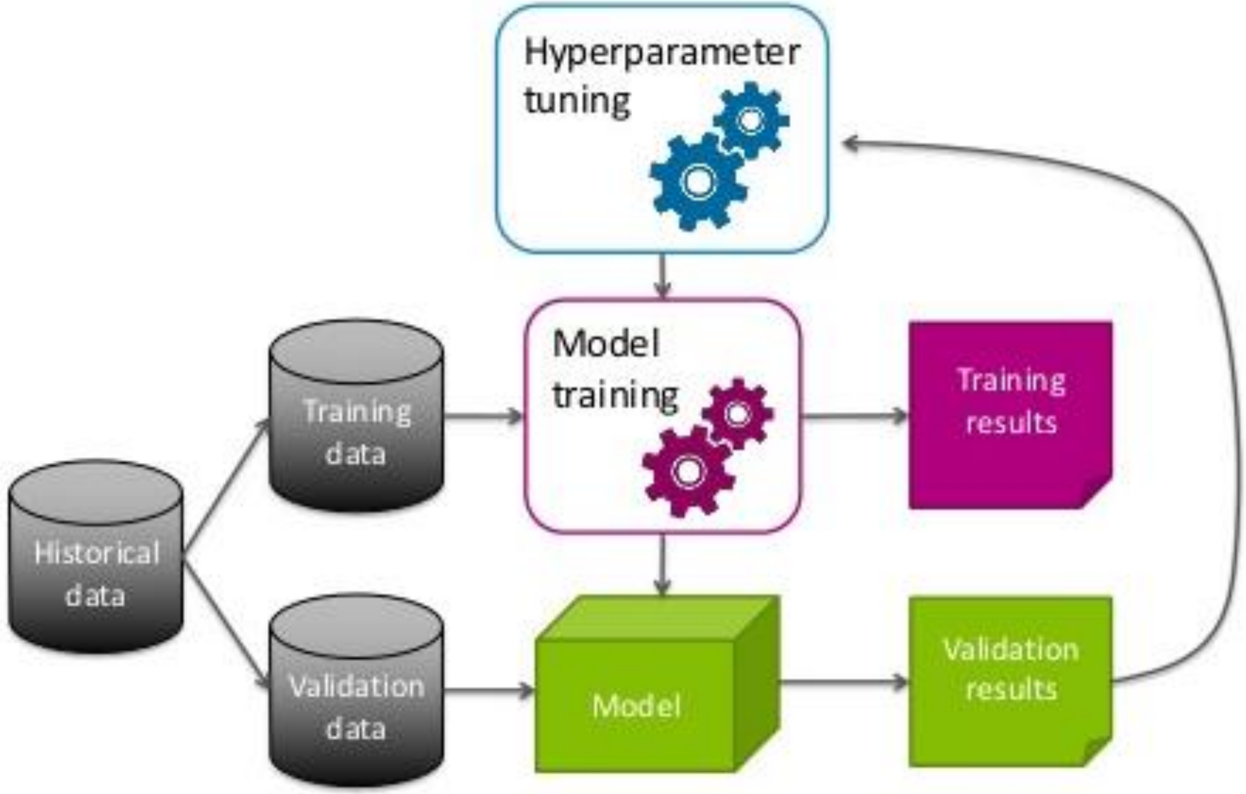
Hyperparameter Tuning/Optimization

- In data mining, **hyperparameter tuning** or **optimization** is the problem of choosing a set of optimal hyperparameters for a learning algorithm.



Validation in Hyperparameter Tuning

- In tuning the hyperparameters, we need to use a set of examples which is different from the one used for training for validation (a hold-out validation set, or k-fold cross validation)



Hyperparameter Tuning in Python

- Use the function *GridSearchCV* in python

```
from sklearn.model_selection import GridSearchCV
parameters={'min_samples_split' : range(5,50,5),'max_depth': range(1,20,1)}

from sklearn.tree import DecisionTreeClassifier
clf_tree = DecisionTreeClassifier(random_state=0)
clf = GridSearchCV(clf_tree,parameters, cv=5)
clf.fit(X_train, y_train)
clf.best_estimator_
```

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=4,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=5,
            min_weight_fraction_leaf=0.0, presort=False, random_state=0,
            splitter='best')
```

Lab: Hyperparameter Tuning

- Files needed
  - Hyperparameter_titanic.ipynb (Python file)
  - titanic_cleaned.csv (dataset)

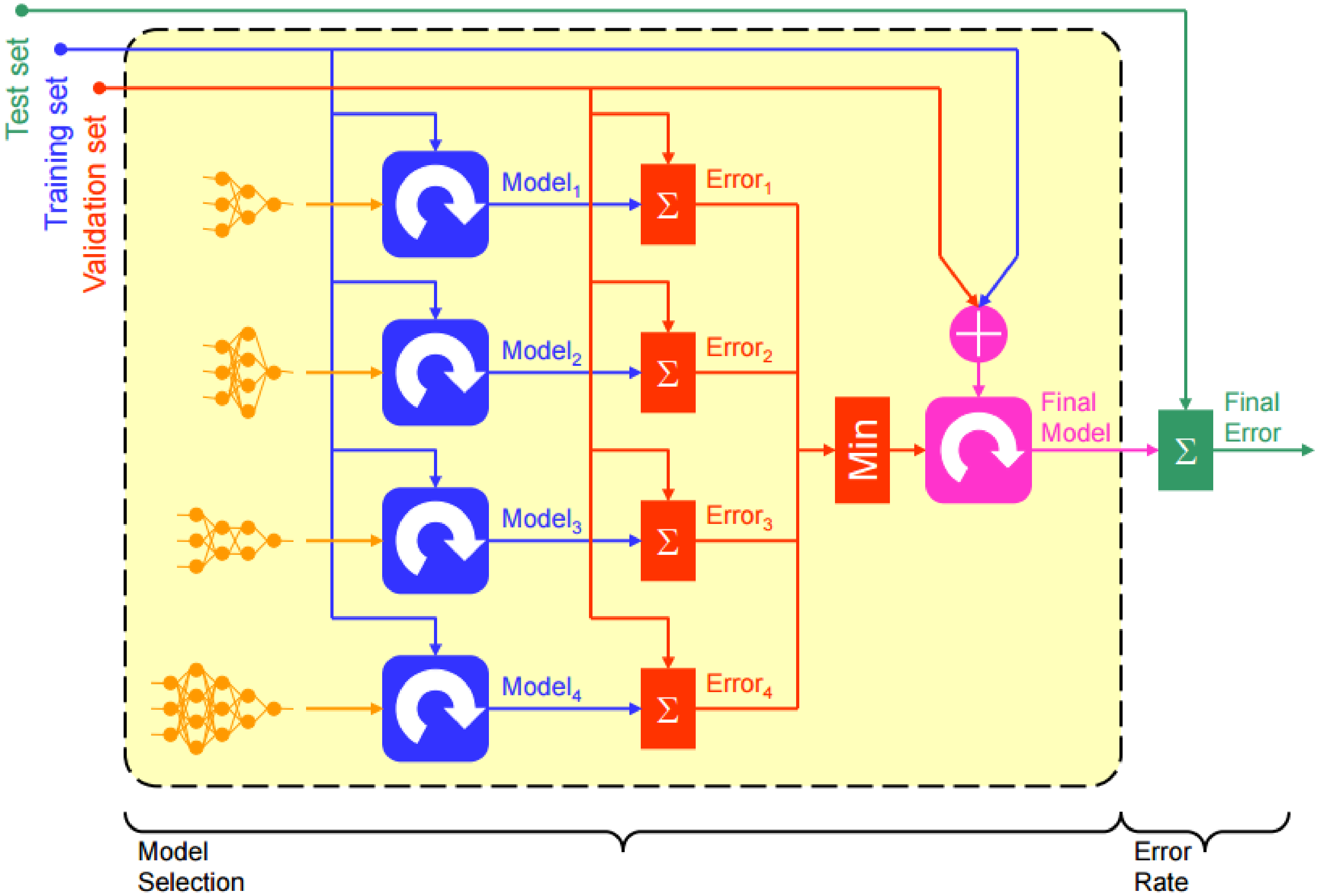
Model Evaluation After Hyperparameter Tuning

- Please note: after hyperparameter tuning, we still need to evaluate the model performance with the optimal set of hyperparameters on test (unseen) examples.
  - Simple holdout validation
  - k-fold cross validation (*nested cross validation*)

Why cannot we simply report the model performance on validation data (previous slide)?

Overfitting

Three-Way Data Splits



Nested Holdout Test (Hyperarameter Tuning)

- Procedure outline
  1) Divide the available data into training, validation and test set
  2) Select induction algorithm and training hyperparameters
  3) Train the model using the training set
  4) Evaluate the model using the validation set
  5) Repeat steps 2) through 4) using different induction algorithms or hyperparameters
  6) Select the best model and train it using data from both training and validation sets
  7) Assess this final model using the test set

--- This outline assumes a holdout method

--- If you want to use k-fold cross validation for hyperparameter tuning (but not testing), steps 3) and 4) need to be repeated for each fold.