Feature Selection in Text

Applied Text Mining

Dr. Maryam Movahedifar

14-17 July 2025

University of Bremen, Germany
movahedm@uni-bremen.de

Universität Bremen

DATA SCIENCE CENTER

Outline

Cross-Validation Method

Data Splitting and Cross-Validation

Why Split Data?

To ensure our models learn well and prove themselves on fresh, unseen data:

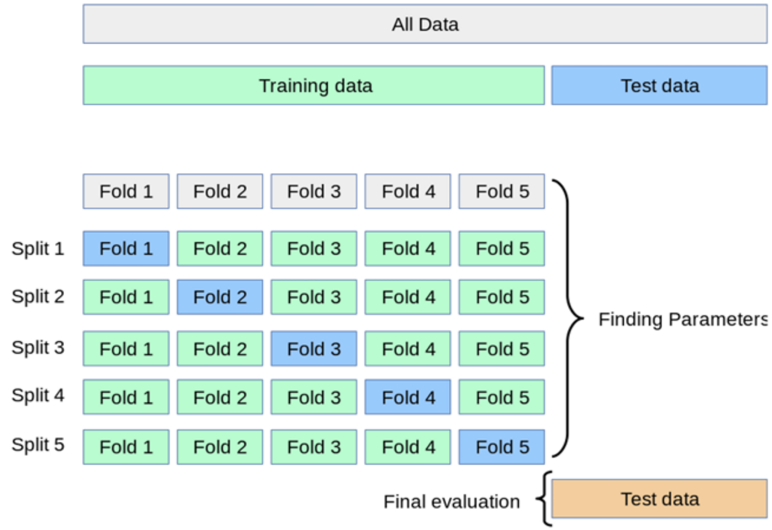
- Training Set: Where the model learns patterns.
- Validation Set: Tune the knobs — hyperparameters.
- Test Set: The final exam — checking real-world readiness.

What is Cross-Validation?

Multiple "mini exams": train and test on different data slices.

This helps avoid overfitting and gives a trustworthy performance estimate.

Cross-Validation: K-Fold



Introduction to Feature Selection

Feature Selection: What and Why

What is Feature Selection?

Selects only the most relevant features from the dataset:

- Reduces dimensionality
- Improves interpretability
- Prevents overfitting

Why Does It Matter?

Crucial for high-dimensional data (e.g., text):

- Cuts computation time
- Increases accuracy
- Focuses on key signals

✓ **Feature Selection simplifies models, sharpens insights, and boosts performance.**

Feature Selection Example

Imagine a dataset with **10,000 features** — like predicting if an email is spam.

- **Goal**: Reduce features to a **manageable size** before modeling.
- You want to cut down from **10,000** to **1,000** features.
- **Question**: Which **1,000 features** should you pick?

This selection process is called Feature Selection.

Why Does Accuracy Reduce with More Features?

Adding more features might seem helpful, but it can actually **reduce accuracy**, even when the original important features are still included.

- Imagine the **best feature set** has **20 features**.
- Now, add 5 extra features — surprisingly, accuracy can drop.
- But wait, why? The original 20 features are still there!

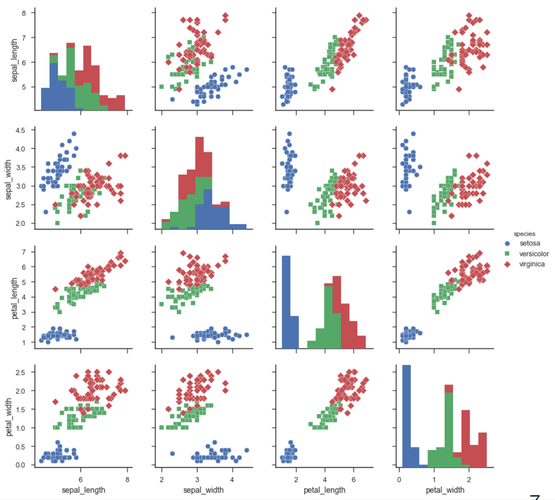
Key Issues:

- Additional features often introduce **noise** that confuses the model.
- They increase **complexity**, making training and optimization harder.
- Extra features raise the risk of **overfitting**, hurting generalization.

Feature Selection Example

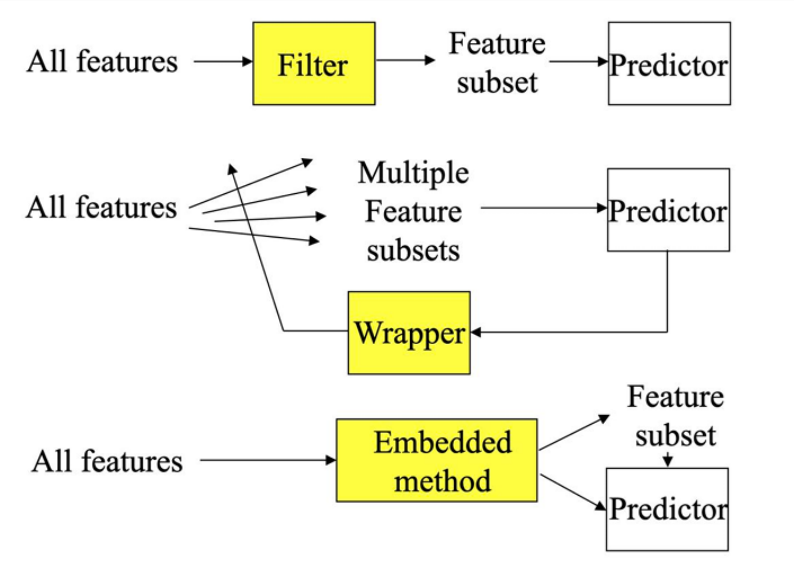
Feature selection is about identifying the most informative features that help distinguish classes in a dataset.

- The plot shows pairs of features illustrating how well they separate the three species: *setosa, versicolor, virginica*.
- Some features, like petal length and petal width, clearly separate the species.
- Others, like sepal length and sepal width, overlap and add less value.
- **Feature selection** keeps only the features that improve classification accuracy.

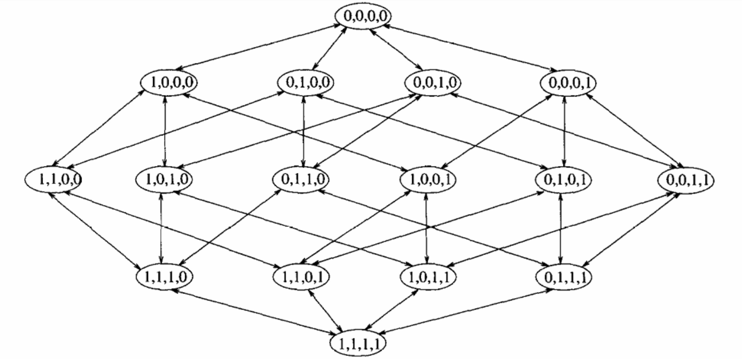


Feature Selection Methods

Feature Selection Methods Overview



Feature Subset Selection: State Space Search



For a dataset with N features, there are 2^N possible subsets of features. Each state represents a feature subset, and the nodes in the search space are connected based on the addition or deletion of a single feature. The search space is too large to exhaustively search for all possible subsets when N is large. Therefore, heuristic search methods are used to guide the search towards the optimal feature subset based on evaluation.

Filter Method

Filter-Based Feature Selection

Filter methods evaluate features **independently of the model** using fast, statistical tests. They are ideal for **large datasets**!

Gini Index: Measures how pure a feature split is — cleaner splits mean better features.

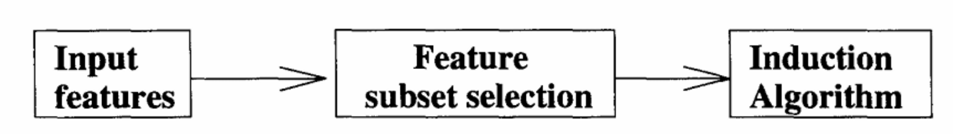
Chi-Square (χ^2): Checks if a feature's distribution depends on the target class.

Mutual Information: Captures how much information a feature shares with the class.

Odds Ratio: Compares the odds of a term appearing with vs. without a class.

Document Frequency: Filters out rare terms that don't help prediction.

Visual Representation of Filter-Based Feature Selection



Key Concept: All features are evaluated independently of the predictive model to identify the most relevant feature subsets.

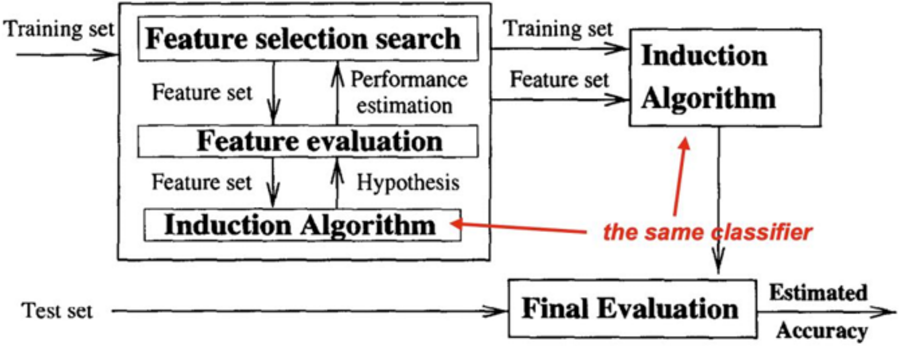
Wrapper Method

Wrapper Methods

Wrapper methods search for the **best subset of features** by evaluating performance on a specific learning algorithm. They train models on different feature subsets and keep the one that performs best.

- Optimized for a specific learning algorithm.
- Feature subsets are evaluated by training and validating models.
- Example: **Recursive Feature Elimination (RFE)**.
- Computationally expensive due to testing many combinations.
- Impractical for large feature spaces or text classification.

Wrapper Approach to Feature Subset Selection



R. Kohavi, G.H. John/Artificial Intelligence 97 (1997) 273-324

Embedded Method

Embedded Method

Embedded methods perform feature selection *during model training*, striking a balance between filter and wrapper approaches.

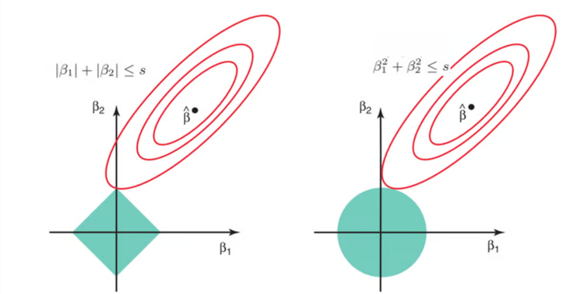
Regularized Learning Objective

$$\min_{\alpha} \hat{F}(\alpha, \sigma) = \sum_{k=1}^{n} L(f(\alpha, \sigma \circ x_k), y_k) + \Omega(\alpha)$$

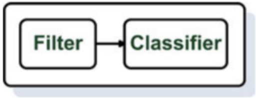
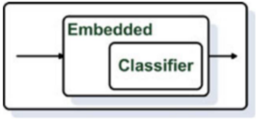
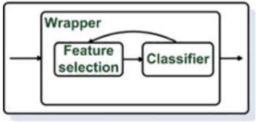
- Feature selection is *built into* the training process.
- **Lasso (L1)**: Shrinks some coefficients to zero $\rightarrow$ *selects features*.
- **Ridge (L2)**: Shrinks all coefficients $\rightarrow$ *keeps all features*.
- More efficient than wrapper methods.

Lasso vs Ridge Regression

- Lasso Regression (L1 Regularization):
    - Shrinks some coefficients to zero.
    - Performs feature selection by removing less important features.
- Ridge Regression (L2 Regularization):
    - Shrinks coefficients but does not set any to zero.
    - Keeps all features in the model, though with reduced weights.



Comparison of Feature Selection Methods

Method	Advantages	Disadvantages
Filter 	Independence of the classifier Lower computational cost than wrappers Fast Good generalization ability	No interaction with the classifier
Embedded 	Interaction with the classifier Lower computational cost than wrappers Captures feature dependencies	Classifier-dependent selection
Wrapper 	Interaction with the classifier Captures feature dependencies	Computationally expensive Risk of overfitting Classifier-dependent selection

Principal Component Analysis (PCA)

What is PCA?

Dimensionality Reduction:

PCA reduces the number of features while retaining as much variance as possible.

Linear Transformation:

Transforms n original features into p uncorrelated components where $p < n$.

Unsupervised Method:

PCA does not consider output labels; it finds structure in the input data alone.

Works Best with Linear Correlations:

It is most effective when features are linearly related.

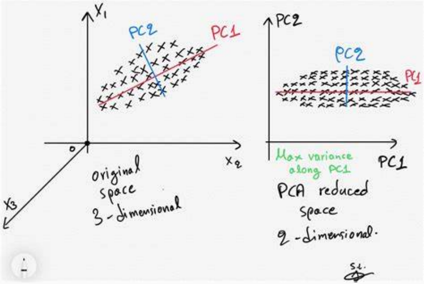
How PCA Works

1. Compute Covariance Matrix
Measure how features vary together to reveal relationships.

2. Perform Eigen Decomposition
Find eigenvectors (directions) and eigenvalues (importance) to capture variance.

3. Project Data
Rotate and project original data onto the new axes defined by the principal components.

4. Select Top p Components
Retain the most informative components to reduce dimensionality.



Feature Selection vs. Feature Reduction

Feature Selection	Feature Reduction
Selects important existing features	Creates new features from combinations of existing ones
Keeps original meaning of features	Transformed features may be harder to interpret
Often used with methods like: Forward Selection, Chi-Square, Lasso	Common methods include: PCA, LDA
Model-specific or independent (depends on method)	Usually model-independent preprocessing
Good for interpretability	Good for compression and noise reduction

Both aim to simplify high-dimensional data, but with different strategies.

Evaluation Metrics Overview

After training and validating a model, it is crucial to evaluate its performance using appropriate metrics. Common evaluation metrics include:

- Accuracy: The ratio of correctly predicted instances to the total instances.
- Precision: Of the predicted positives, how many are actually correct?
- Recall: Of the actual positives, how many were correctly predicted?
- F1-Score: A harmonic mean of precision and recall, balancing both metrics.

Confusion Matrix

A Confusion Matrix is a table used to evaluate the performance of a classification model by comparing actual vs. predicted labels:

- **True Positives (TP):** Correctly predicted positive instances.
- **False Positives (FP):** Incorrectly predicted positive instances.
- **True Negatives (TN):** Correctly predicted negative instances.
- **False Negatives (FN):** Incorrectly predicted negative instances.

Confusion Matrix Example

		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Conclusion

Conclusion

- **Importance:** Feature selection is vital for creating **efficient** and **interpretable** models in text mining.
- **Variety:** Use **Filter**, **Wrapper**, and **Embedded** methods depending on your task and data.
- **Strategy:** Choose a method that balances **performance**, **interpretability**, and **computational cost**.

Good feature selection is the foundation of smart machine learning.

Programming

Practical 3