Chapter-4 (Machine Learning Concept 1)

Basics of Feature Engineering

In this chapter, we study one of the most important aspects of machine learning - **Feature Engineering**. It focuses on three key components:

- Feature Construction
- Feature Selection
- Feature Transformation

Introduction

- Machine Learning requires preparatory steps before modeling.
- One of the critical preparatory steps is **Feature Engineering**.
- It transforms raw input data into meaningful, well-aligned features ready to be used by ML models.

What is a Feature?

- A **feature** is an attribute of a dataset used in a machine learning process.
- Features are also called **dimensions**.
- Example: The famous **Iris dataset** has features: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species (class variable).

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
6.7	3.3	5.7	2.5	Virginica
4.9	3	1.4	0.2	Setosa
5.5	2.6	4.4	1.2	Versicolor
6.8	3.2	5.9	2.3	Virginica
5.5	2.5	4	1.3	Versicolor
5.1	3.5	1.4	0.2	Setosa
6.1	3	4.6	1.4	versicolor

FIG. 4.1 Data set features

What is Feature Engineering?

- The process of translating a dataset into features such that they represent the dataset more effectively and improve learning performance.
- Two major elements:
  1. Feature Transformation
  2. Feature Subset Selection

1. Feature Transformation

- Purpose:
  - Dimensionality reduction
  - Improve learning efficiency
- two approaches:
  1. Feature Construction (add new features)
  2. Feature Extraction (derive new features from existing ones)

1.1 Feature Construction

- Feature construction involves transforming a given set of input features to generate a new set of more powerful features.

Example: Let's take the example of a real estate data set having details of all apartments sold in a specific region

apartment_ length	apartment_ breadth	apartment_ price
80	59	23,60,000
54	45	12,15,000
78	56	21,84,000
63	63	19,84,000
83	74	30,71,000
92	86	39,56,000

apartment_ length	apartment_ breadth	apartment_ area	apartment_ price
80	59	4,720	23,60,000
54	45	2,430	12,15,000
78	56	4,368	21,84,000
63	63	3,969	19,84,500
83	74	6,142	30,71,000
92	86	7912	39,56,000

FIG. 4.2 Feature construction (example 1)

- Instead of using length and breadth, construct a new feature = apartment area (length $\times$ breadth).
- This transforms the dataset from 3D to 4D.

Situations requiring feature construction:

- when features have categorical value and machine learning needs numeric value inputs
- when features having numeric (continuous) values and need to be converted to ordinal values
- when text-specific feature construction needs to be done

Encoding categorical (nominal) variables

- Example: Athlete dataset with features → Age, City of origin, Parents athlete, Chance of win.
- ML algorithms require numeric values.
- Create dummy features for categorical variables.
  - City of origin (A, B, C) → origin_city_A, origin_city_B, origin_city_C.
  - Parents athlete (Y/N) → parents_athlete_Y, parents_athlete_N.
  - Chance of win (Y/N) → win_chance_Y, win_chance_N.
- Each dummy variable gets 0/1 values.
- FIG. 4.3 Feature construction (encoding nominal variables).
- Optimization: If a categorical variable has only two values, one dummy can be dropped to avoid duplication.
- FIG. 4.3c Optimized dummy variable construction.

Age (Years)	City of origin	Parents athlete	Chance of win
18	City A	Yes	Y
20	City B	No	Y
23	City B	Yes	Y
19	City A	No	N
18	City C	Yes	N
22	City B	Yes	Y

(a)

Age (Years)	origin_ city_A	origin_ city_B	origin_ city_C	parents_ athlete_Y	parents_ athlete_N	win_ chance_Y	win_ chance_N
18	1	0	0	1	0	1	0
20	0	1	0	0	1	1	0
23	0	1	0	1	0	1	0
19	1	0	0	0	1	0	1
18	0	0	1	1	0	0	1
22	0	1	0	1	0	1	0

(b)

Age (Years)	origin_city_A	origin_city_B	origin_city_C	parents_athlete_Y	win_chance_Y
18	1	0	0	1	1
20	0	1	0	0	1
23	0	1	0	1	1
19	1	0	0	0	0
18	0	0	1	1	0
22	0	1	0	1	1

(c)

FIG. 4.3 Feature construction (encoding nominal variables)

Encoding categorical (ordinal) variables

- Example: Student dataset → science marks, maths marks, grade.
- Grade (A, B, C, D) is ordinal.
- Create numeric feature num_grade:
  A = 1, B = 2, C = 3, D = 4.

marks_science	marks_maths	Grade
78	75	B
56	62	C
87	90	A
91	95	A
45	42	D
62	57	B

(a)

marks_science	marks_maths	num_grade
78	75	2
56	62	3
87	90	1
91	95	1
45	42	4
62	57	2

(b)

FIG. 4.4 Feature construction (encoding ordinal variables)

Transforming numeric (continuous) features to categorical

- Sometimes continuous values are binned into categories.
- Example: Apartment_price (continuous) → Price-grade (categorical).
- Enables converting regression task → classification task.

apartment_ area	apartment_ price
4,720	23,60,000
2,430	12,15,000
4,368	21,84,000
3,969	19,84,500
6,142	30,71,000
7,912	39,56,000

(a)

apartment_ area	apartment_ grade
4,720	Medium
2,430	Low
4,368	Medium
3,969	Low
6,142	High
7,912	High

(b)

apartment_ area	apartment_ grade
4,720	2
2,430	1
4,368	2
3,969	1
6,142	3
7,912	3

(c)

FIG. 4.5 Feature construction (numeric to categorical)

Text-specific feature construction

- Text is predominant in communication (social media, email, messaging).
- Text = **unstructured**, needs conversion to numeric features.
- Process = **Vectorization** (Bag-of-Words model).
  1. **Tokenize** → split into words/tokens.
  2. **Count** → word occurrences.
  3. **Normalize** → weight terms (reduce importance of common words).
- Creates a document-term matrix (rows = documents, columns = tokens, values = counts).

This	House	Build	Feeling	Well	Theatre	Movie	Good	Lonely	...
2	1	1	0	0	1	1	1	0	
0	0	0	1	1	0	0	0	0	
1	0	0	2	1	1	0	0	1	
0	0	0	0	1	0	1	1	0	
.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	

FIG. 4.6 Feature construction (text-specific)

1.2. Feature extraction

Feature extraction creates new features from a combination of original ones.

- Operators for combining features:
  - **Boolean:** Conjunctions, Disjunctions, Negation.
  - **Nominal:** Cartesian product, M-of-N, etc.
  - **Numerical:** Min, Max, Addition, Subtraction, Multiplication, Division, Average, Inequality.

Example:

- Dataset feature set $F_i(F_1, F_2, \ldots, F_n)$.
- After extraction $\rightarrow$ new set $G_i(G_1, G_2, \ldots, G_m)$ such that G_i=f (F_i) where $m < n$.
- Example: G_1=f(F_1,F_2).

Feat_A	Feat_B	Feat_C	Feat_D
34	34.5	23	233
44	45.56	11	3.44
78	22.59	21	4.5
22	65.22	11	322.3
22	33.8	355	45.2
11	122.32	63	23.2

Feat_1	Feat_2
41.25	185.80
54.20	53.12
43.73	35.79
65.30	264.10
37.02	238.42
113.39	167.74

$$\text{Feat}_1 = 0.3 \times \text{Feat}_A + 0.9 \times \text{Feat}_A$$
$$\text{Feat}_2 = \text{Feat}_A + 0.5\,\text{Feat}_B + 0.6 \times \text{Feat}_C$$

Let's discuss the most popular feature extraction algorithms used in machine learning:

Principal Component Analysis

- Many dataset features are correlated (e.g., Height $\Leftrightarrow$ Weight).
- Goal: reduce correlated features $\rightarrow$ fewer, independent features.
- PCA transforms n-dimensional feature space into m-dimensional space (orthogonal features).
- **Principal components** = new uncorrelated features capturing variance.

Key objectives:

1. New features (principal components) have zero covariance.
2. Components ordered by variance captured (first captures max variance).
3. Total variance preserved.

Steps:

- Compute covariance matrix.
- Calculate eigenvalues.
- Eigenvector with largest eigenvalue = 1st principal component.
- Next eigenvector (orthogonal) = 2nd principal component.
- Select top k eigenvectors $\rightarrow$ k principal components.

Singular Value Decomposition (SVD)

- A matrix factorization technique.
- Decomposes matrix A (m × n):
  - $A = U_{(m \times m)} \sum_{(m \times n)} V^T_{(n \times n)}$

    where, U and V are orthonormal matrices, $\sum$ is rectangular diagonal matrix.
  - U = left singular vectors, V = right singular vectors, $\sum$ = singular values.
- Properties:
  1. Patterns in attributes → right singular vectors (V).
  2. Patterns in instances → left singular vectors (U).
  3. Larger singular value → larger contribution of matrix A.
- New k-dimensional matrix = D × $[V_1, V_2, \ldots, V_k]$.
- Good for sparse data (e.g., text).

Linear Discriminant Analysis (LDA)

- Like PCA: reduces dimensionality.
- Unlike PCA: focuses on **class separability**, not variance.
- Prevents overfitting by maximizing discrimination between classes.

Steps:

1. Compute mean vectors for each class.
2. Compute intra-class (Sw) and inter-class (Sb) scatter matrices.
3. Calculate eigenvalues/eigenvectors for $S_w^{-1} S_b$.
4. Select top k eigenvectors $\rightarrow$ project dataset into reduced space.

2. Feature Subset Selection

Feature selection = most critical pre-processing activity in ML.

- **Aim:** choose a subset of features that contribute meaningfully.
- **Example:** Student weight prediction dataset → Roll Number, Age, Height, Weight.
  - Roll Number irrelevant → remove it.
  - Subset = {Age, Height, Weight}.
- Expected to improve results.

Roll Number	Age	Height	Weight
12	12	1.1	23
14	11	1.05	21.6
19	13	1.2	24.7
32	11	1.07	21.3
38	14	1.24	25.2
45	12	1.12	23.4

→

Age	Height	Weight
12	1.1	23
11	1.05	21.6
13	1.2	24.7
11	1.07	21.3
14	1.24	25.2
12	1.12	23.4

FIG. 4.8 Feature selection

Let's try to understand the issues which have made feature selection such a relevant problem to be solved

Issues in High-Dimensional Data

- Data growth → huge, high-dimensional datasets.
- Domains: DNA analysis, GIS, social networking.
- Example: DNA microarray → up to 450,000 variables.
- **Problems:**
  - High computational resources needed.
  - Noise degrades performance.
  - Hard to interpret models.
- **Solution:** Select subset of features.

Objectives of feature selection:

1. Faster, cost-effective learning (less computation).
2. Improved efficiency of model.
3. Better understanding of underlying data model.

Key Drivers – Feature Relevance & Redundancy

Feature Relevance

- **Supervised learning:** predictor variables contribute to class labels.
  - Irrelevant → no info.
  - Weakly relevant → small info contribution.
  - Strongly relevant → significant info contribution.
- **Unsupervised learning:** grouping by similarity.
  - Irrelevant features → no role in grouping.
- Example: Student Roll Number irrelevant for predicting weight or grouping by merit.

Feature Redundancy

- Feature adds similar info as another feature.
- Example: Weight prediction → both Age & Height correlated with Weight.
- Redundant features → can drop one, keep representative subset.

Goal: remove irrelevant + redundant features → meaningful subset.

Measures of Feature Relevance & Redundancy

Measures of Feature Relevance

- Based on information contribution.
- Supervised: Mutual Information (MI) between feature and class. Mutual information can be calculated as follows:

$$MI(C, f) = H(C) + H(f) - H(C, f)$$

where, marginal entropy of the class,

$H(C) = -\sum_{i=1}^{k} p(C_i) \log_2 p(C_i)$

marginal entropy of the feature 'x',

$H(f) = -\sum_{c} p(f = x) \log_2 p(f = x)$

and K = number of classes, C = class variable, f = feature set that take discrete values.

- Unsupervised: entropy-based feature ranking (Shannon's entropy).

The entropy of a feature f is calculated using Shannon's formula below:

$$H(f) = -\sum_{x} p(f = x) \log_2 p(f = x)$$

$\sum_{x}$ is used only for features that take discrete values.

Measures of Feature Redundancy

- Based on similarity of info.

1. Correlation-based

Correlation is a measure of linear dependency between two random variables. For two random feature variables F_1 and F_2 , Pearson correlation coefficient is defined as:
Let α be the correlation coefficient between two datasets F_1 and F_2. Then:

$$\alpha = \frac{\text{cov}(F_1, F_2)}{\sqrt{\text{var}(F_1) \cdot \text{var}(F_2)}}$$

where the covariance is defined as:

$$\text{cov}(F_1, F_2) = \sum_{i=1}^{n} (F_{1i} - \bar{F}_1)(F_{2i} - \bar{F}_2)$$

and the variances are:

$$\text{var}(F_1) = \sum_{i=1}^{n}(F_{1i} - \bar{F}_1)^2, \quad \text{where } \bar{F}_1 = \frac{1}{n}\sum_{i=1}^{n} F_{1i}$$

$$\text{var}(F_2) = \sum_{i=1}^{n}(F_{2i} - \bar{F}_2)^2, \quad \text{where } \bar{F}_2 = \frac{1}{n}\sum_{i=1}^{n} F_{2i}$$

Correlation values range between -1 and $+1$. Specifically:

- A correlation of $+1$ implies a perfect positive linear relationship.
- A correlation of -1 implies a perfect negative linear relationship.
- A correlation of 0 suggests **no linear relationship** between the features.

In the context of **feature selection**, correlation is often used to assess the similarity or redundancy between features. A threshold value is typically adopted to determine whether two features are sufficiently similar.

2. Distance-based

The most common distance measure is the Euclidean distance, which, between two features F_1 and F_2 are calculated as:

$$d(F_1, F_2) = \sqrt{\sum_{i=1}^{n}(F_{1i} - F_{2i})^2}$$

where F_1 and F_2 are features of an n-dimensional data set.

Aptitude (F_1)	Communication (F_2)	($F_1 - F_2$)	($F_1 - F_2$)^2
2	6	–4	16
3	5.5	–2.5	6.25
6	4	2	4
7	2.5	4.5	20.25
8	3	5	25
6	5.5	0.5	0.25
6	7	–1	1
7	6	1	1
8	6	2	4
9	7	2	4
			81.75

FIG. 4.9 Distance calculation between features

A more generalized form of the Euclidean distance is the **Minkowski** distance, measured as

$$d(F_1, F_2) = \left(\sum_{i=1}^{n} (F_{1i} - F_{2i})^r \right)^{\frac{1}{r}}$$

- r=2 $\rightarrow$ Euclidean
- r=1 $\rightarrow$ Manhattan
- Special case: Hamming distance (binary vectors).
  For example, the Hamming distance between two vectors 01101011 and 11001001 is 3.

Overall Feature Selection Process

Feature selection is the process of selecting a subset of features in a data set.

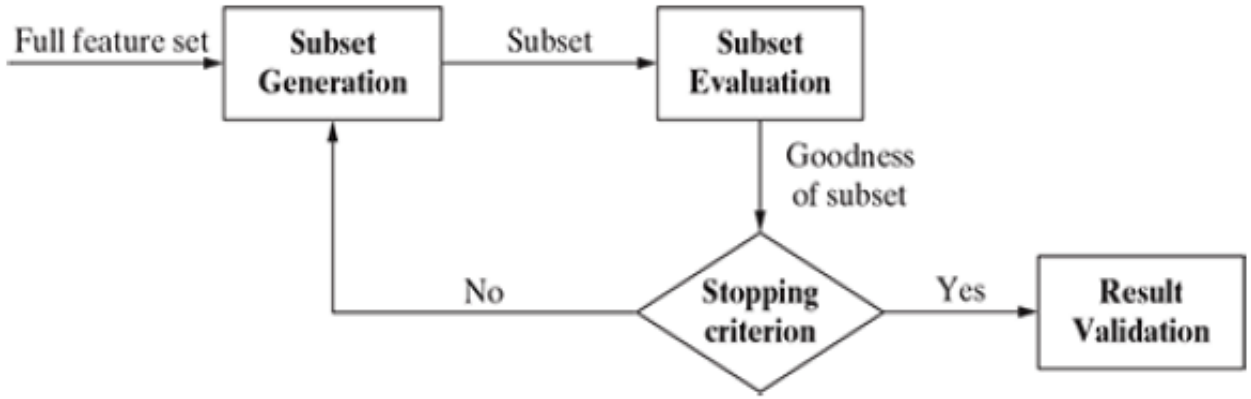

FIG. 4.12 Feature selection process

1. **Generation of possible subsets**
   - n features $\rightarrow 2^n$ subsets (impractical).
   - Use approximate strategies:
     - Sequential forward selection (start empty $\rightarrow$ add).
     - Sequential backward elimination (start full $\rightarrow$ remove).
     - Bi-directional (both ends).
2. Subset evaluation $\rightarrow$ compare with best subset.
3. Stop criterion $\rightarrow$ e.g. iteration bound, no improvement, or good enough subset.
4. Validation $\rightarrow$ test with benchmarks or real datasets.

- Supervised: accuracy.
- Unsupervised: cluster quality.

Feature Selection Approaches

There are four types of approach for feature selection:

1. **Filter Approach** (FIG. 4.13 Filter approach)
   - Uses statistical measures only, no ML model.
   - Examples: correlation, info gain, Fisher score, ANOVA, Chi-Square.

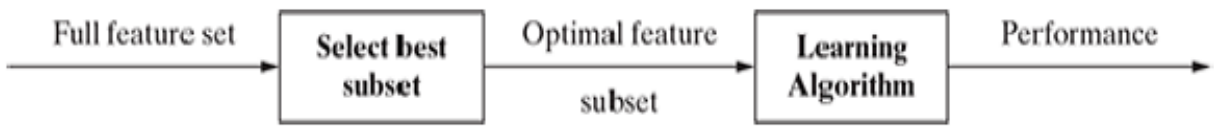

FIG. 4.13 Filter approach

2. **Wrapper Approach** (FIG. 4.14 Wrapper approach)
   - Uses induction algorithm as black box.
   - For each candidate subset → train ML model & evaluate.
   - Computationally expensive but better performance.

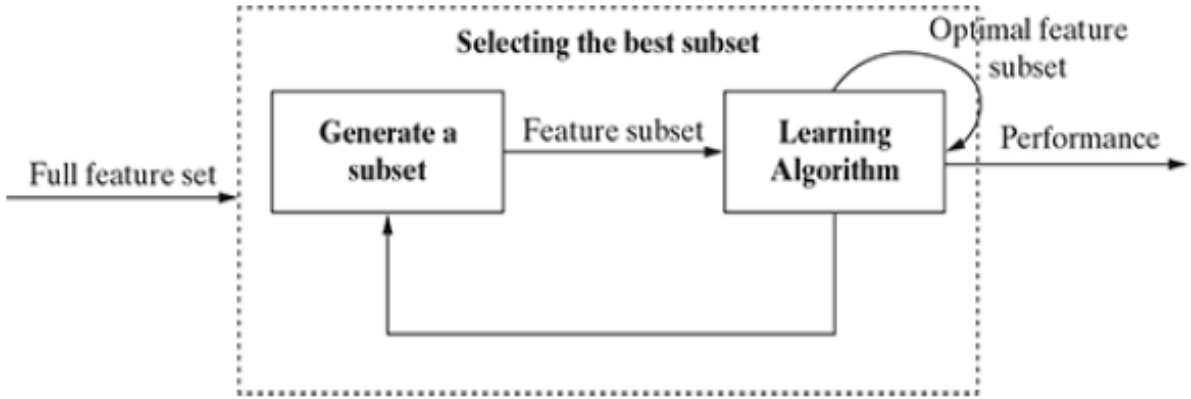

FIG. 4.14 Wrapper approach

3. **Hybrid Approach**
   - Combines filter + wrapper.
   - Filter to shortlist subsets → wrapper to finalize.
4. **Embedded Approach** (FIG. 4.15 Embedded approach)
   - Feature selection + model training happen simultaneously.

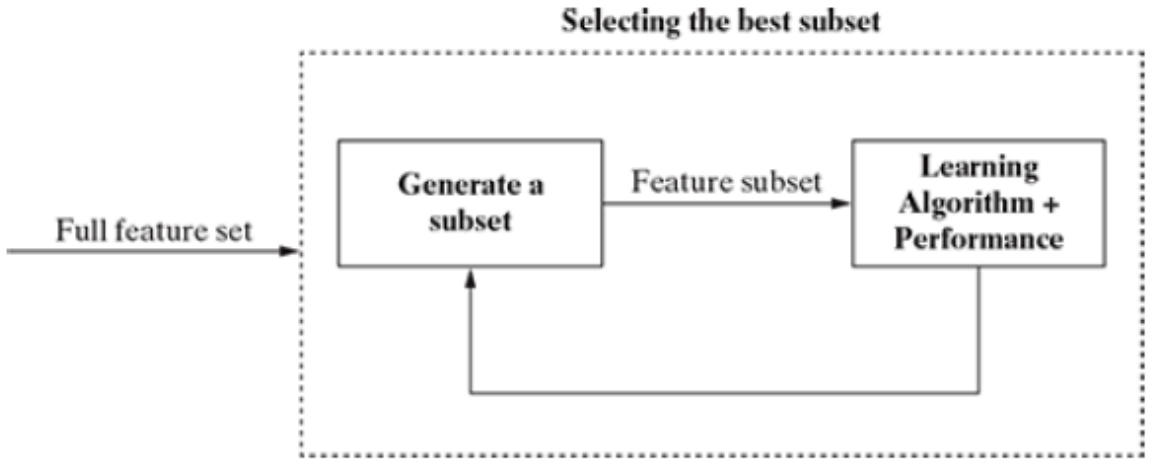

FIG. 4.15 Embedded approach