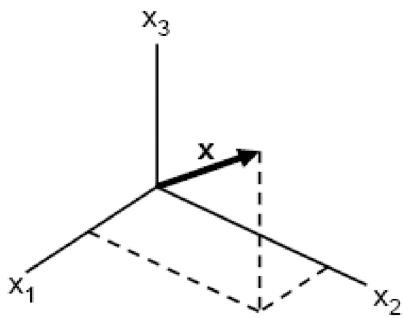
Feature Engineering

Prof. Dr. M. Elif Karslıgil
Yıldız Teknik Üniversitesi
Computer Engineering Department
Intelligent Systems Lab.

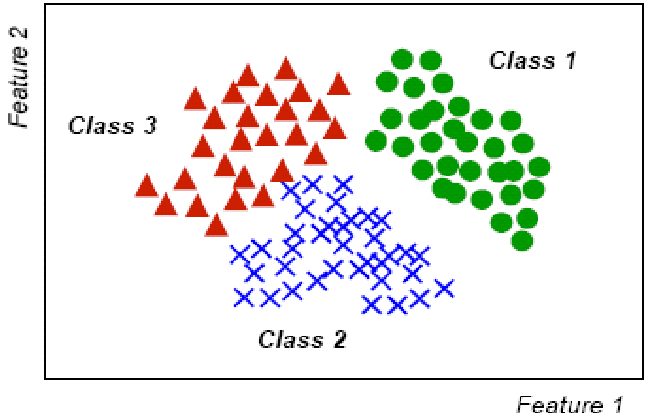
What is Feature ?

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

Feature vector

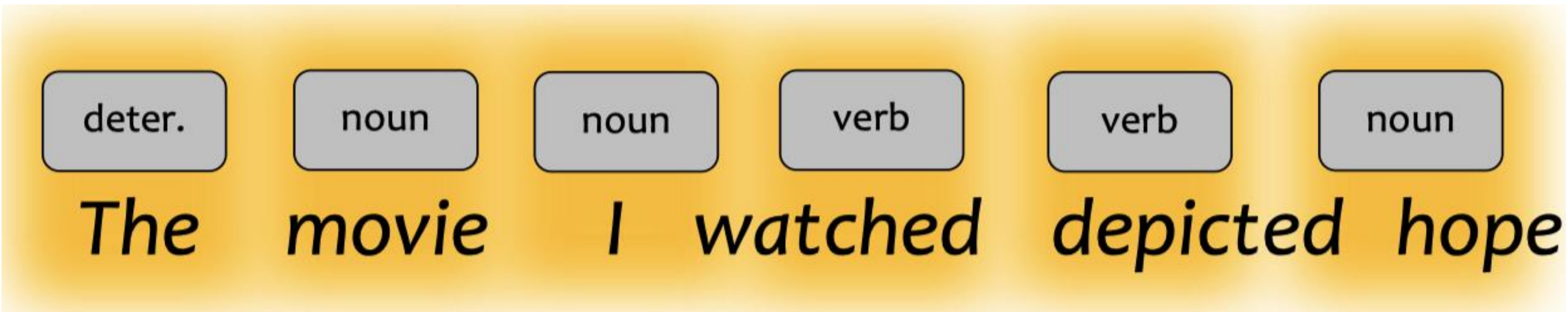


Feature space (3D)

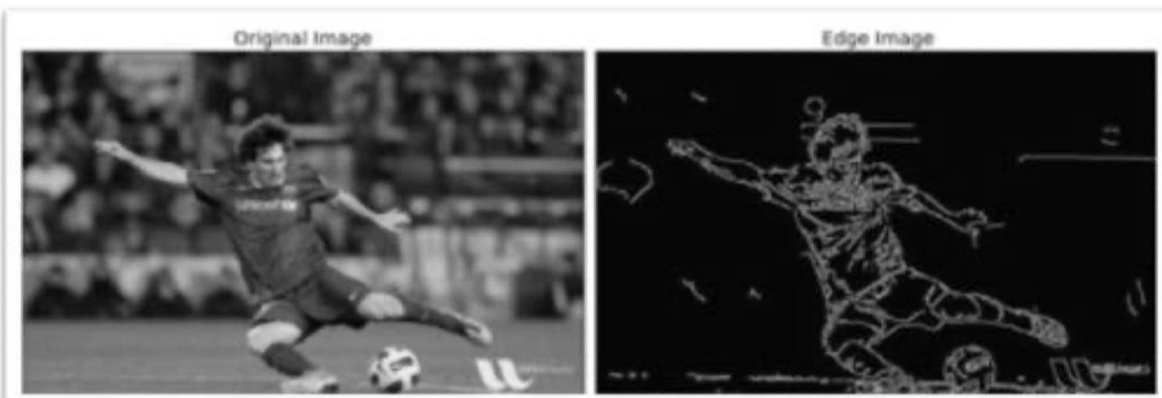


Scatter plot (2D)

Hand Crafted Features



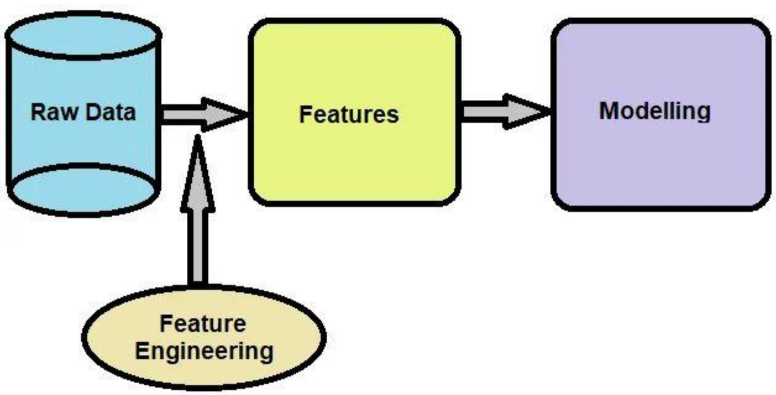
Edge detection (Canny)



What is Feature Engineering?

- Feature engineering is transforming raw data into informative variables (features)

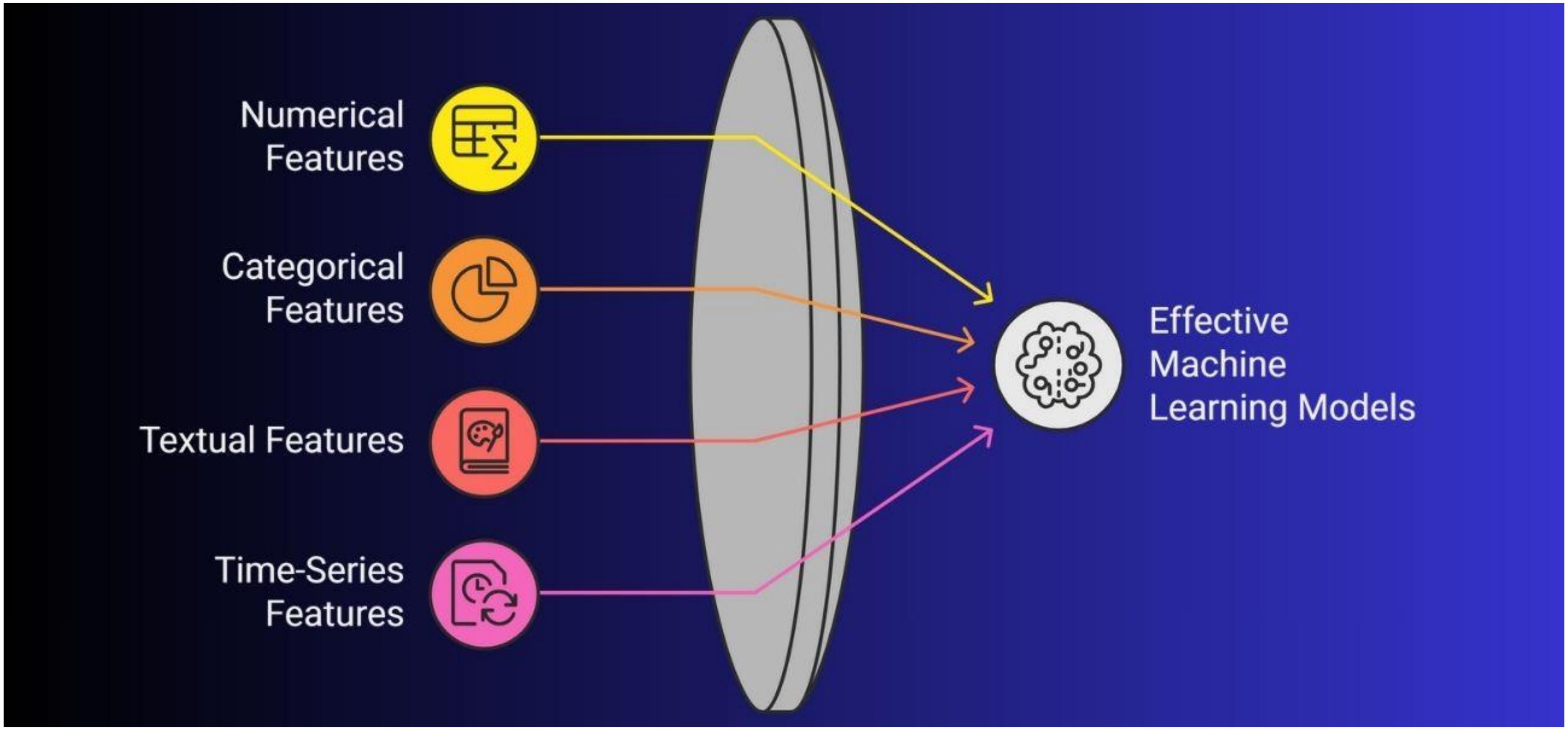
- It is the process of creating, transforming, or selecting input variables (features) so that a machine learning model can learn patterns more easily and effectively



Why Feature Engineering ?

1. Improves Accuracy: It helps the model capture complex relationships that raw data might hide
2. Enhances Interpretability: Clear, meaningful features make it easier to understand how a model makes decisions.
3. Reduces Overfitting: Focusing on important features helps models generalize better to new, unseen data.
4. Boosts Efficiency: Using fewer but high-impact features speeds up model training and reduces computational cost.

Types of Features



Categorical Features

Categorical feature engineering converts text-based categories into meaningful numerical values using encoding methods like one-hot, label, and target encoding.

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

The problem with one-hot encoding is that it increases the dimensionality of the training data

Categorical Features

- Target Encoding:
  - Each category is encoded by how the target behaves for that cate

- X = categorical feature
- y = target variable

For category c:

$$\text{TargetEncode}(c) = \mathbb{E}[y \mid X = c]$$

For binary classification:

$$\text{TargetEncode}(c) = P(y = 1 \mid X = c)$$

Categorical Features

- Target Encoding Example

	Animal	Target	Encoded Animal
0	cat	1	0.40
1	hamster	0	0.50
2	cat	0	0.40
3	cat	1	0.40
4	dog	1	0.67
5	hamster	1	0.50
6	cat	0	0.40
7	dog	1	0.67
8	cat	0	0.40
9	dog	0	0.67

	Animal Group	Target 0	Target 1	Probability of 1
0	cat	3	2	0.40
1	dog	1	2	0.67
2	hamster	1	1	0.50

Textual Features

1. Bag of Words (BoW) Example :

- **Review 1:** This movie is very scary and long
- **Review 2:** This movie is not scary and is slow
- **Review 3:** This movie is spooky and good

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

Vector-of-Review1: [1 1 1 1 1 1 1 0 0 0 0]

Vector-of-Review2: [1 1 2 0 0 1 1 0 1 0 0]

Vector-of-Review3: [1 1 1 0 0 0 1 0 0 1 1]

Textual Features

1. Drawbacks of Bag of Words (BoW):

- If the new sentences contain new words, then our vocabulary size would increase and thereby, the length of the vectors would increase too.
- Additionally, the vectors would also contain many 0s, thereby resulting in a sparse matrix (which is what we would like to avoid)
- We are retaining no information on the grammar of the sentences nor on the ordering of the words in the text.

Textual Features

2. TF-IDF (Term Frequency — Inverse Document Frequency:

- Weights words by:
  - how frequent they are in the document
  - how rare they are across documents

$$TF = \frac{Number\ of\ times\ a\ word\ "X"\ appears\ in\ a\ Document}{Number\ of\ words\ present\ in\ a\ Document}$$

$$IDF = log\left(\frac{Number\ of\ Documents\ present\ in\ a\ Corpus}{Number\ of\ Documents\ where\ word\ "X"\ has\ appeared}\right)$$

$$TF\ IDF\ =\ TF * IDF$$

Term Frequency *:* how often a term appears in a document

Inverse Document Frequency*:* how rare or unique the item is across the entire collection of document.

Textual Features

2. TF-IDF (Term Frequency — Inverse Document Frequency:

- Weights words by:
  - how frequent they are in the document
  - how rare they are across documents

$$TF = \frac{Number\ of\ times\ a\ word\ "X"\ appears\ in\ a\ Document}{Number\ of\ words\ present\ in\ a\ Document}$$

$$IDF = log\left(\frac{Number\ of\ Documents\ present\ in\ a\ Corpus}{Number\ of\ Documents\ where\ word\ "X"\ has\ appeared}\right)$$

$$TF\ IDF\ =\ TF * IDF$$

Term Frequency *:* how often a term appears in a document

Inverse Document Frequency*:* how rare or unique the item is across the entire collection of document.

Textual Features

3. Word Embeddings (Dense semantic features)

- A word embedding is a learned representation for text where words that have the same meaning have a similar representation

- Words become points in a semantic space

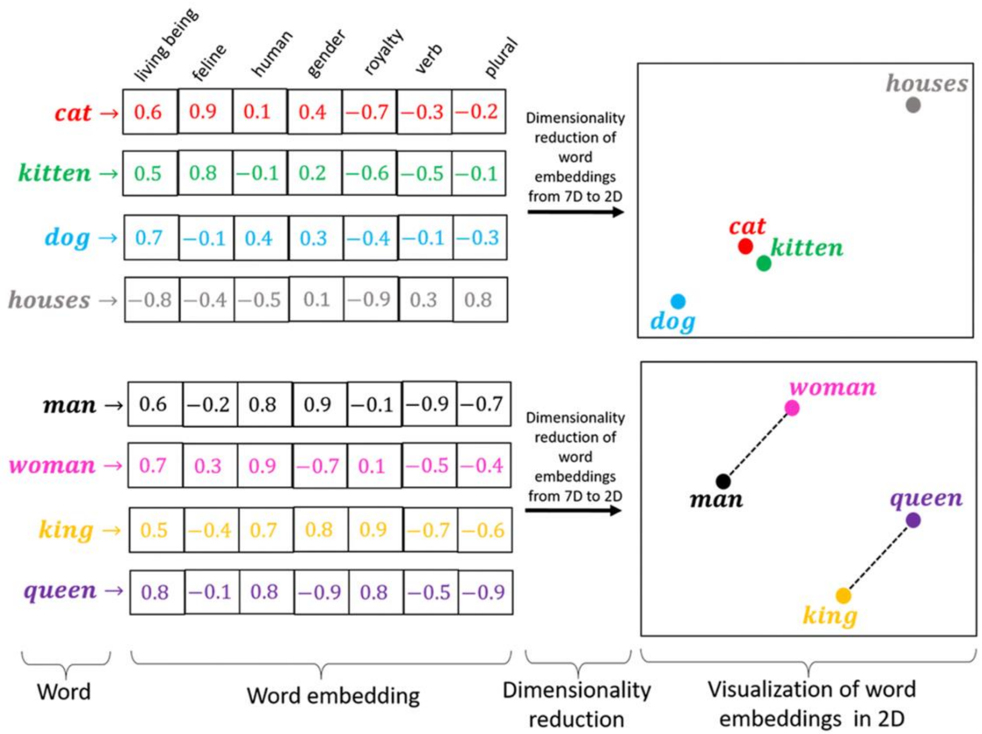
- An embedding is a dense vector of floating point values (the length of the vector is a parameter you specify).

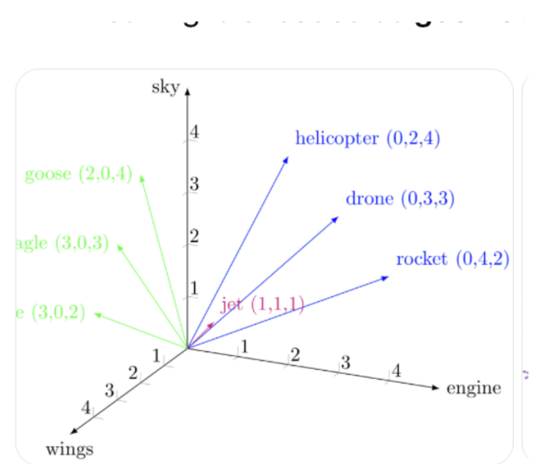
- Map each word to a **dense vector** where:
  –similar words → close vectors
  –meaning is encoded geometricall

- Instead of specifying the values for the embedding manually, they are trainable parameters (weights learned by the model during training, in the same way a model learns weights for a dense layer)

Textual Features

3. Word Embeddings (Dense semantic features)



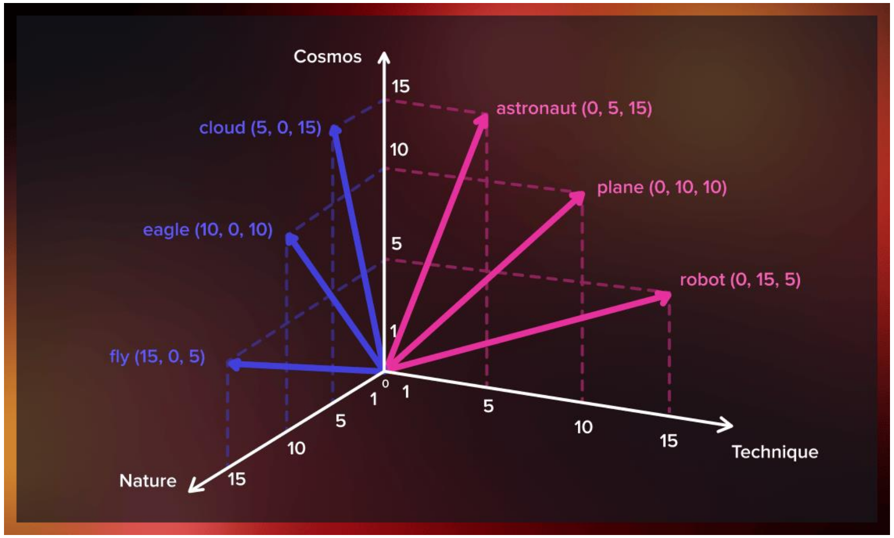


Textual Features

3. Word Embeddings (Dense semantic features)

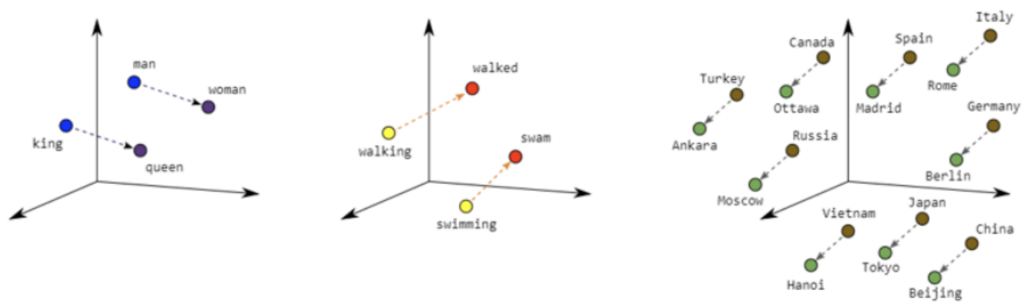
Word2Vec:

- Words that appear in similar contexts get similar vectors
- It is trained using a neural network that learns the relationships between words in large databases of texts
- It learns word embeddings by predicting surrounding words,turning context similarity into geometric similarity.
- Similar words have similar vectors



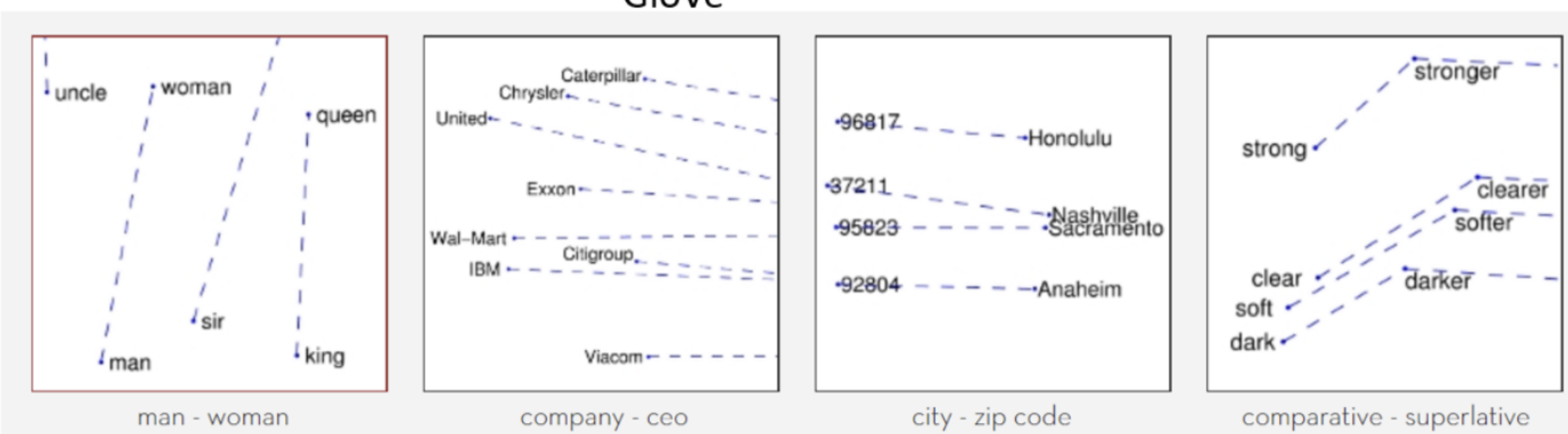
Textual Features

Word2Vec



Male-Female Verb Tense Country-Capital

GloVe

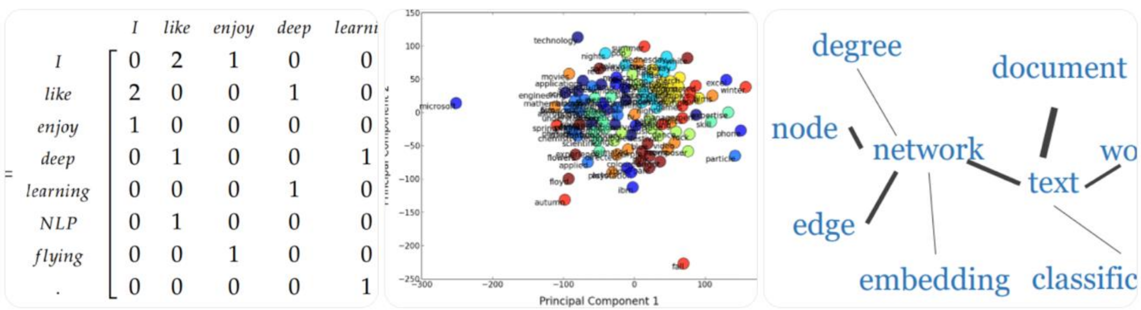


Textual Features

3. Word Embeddings (Dense semantic features)

GloVe (Global Vector for word Representation)

- learns word embeddings by using global word co-occurrence statistics from a corpus.
- Word meaning is encoded in ratios of co-occurrence probabilities(ice co-occurs more with snow).
- GloVe learns vectors with self supervised regression



	I	like	enjoy	deep	learni
I	0	2	1	0	0
like	2	0	0	1	0
enjoy	1	0	0	0	0
deep	0	1	0	0	1
learning	0	0	0	1	0
NLP	0	1	0	0	0
flying	0	0	1	0	0
.	0	0	0	0	1

X_{ij} number of times word j appears in the context of word I
Context is defined using sliding window and distance weighting

Applications of Word2Vec and GloVe

- text classification, similarity search,
- recommendation systems,
- Information retrieval
- Document clustering

Choose Word2Vec if:

- you train on your own corpus
- data is domain-specific
- memory is limited
- you want fast iteration

Choose GloVe if:

- you use pretrained embeddings
- you want strong semantic regularities
- corpus is static and large

Time Series Features

Time series features are numerical descriptors extracted from sequential temporal data

1. **Statistical Features:**

- **Ignores time order**, summarize **value distribution**.
- Useful when magnitude matters more than shape
- Applications: medicine(heart rate variability), finance(high variance shows unstable system behavior)
- Features: **mean, median, variance, standard deviation, kurtosis etc.**
- Not enough for: long term depencies, forecasting future values, phase and ordering is critical
- Can’t use for: speech recognition, language modelling, gesture recognition

Time Series Features

2. **Frequency Domain Features:**

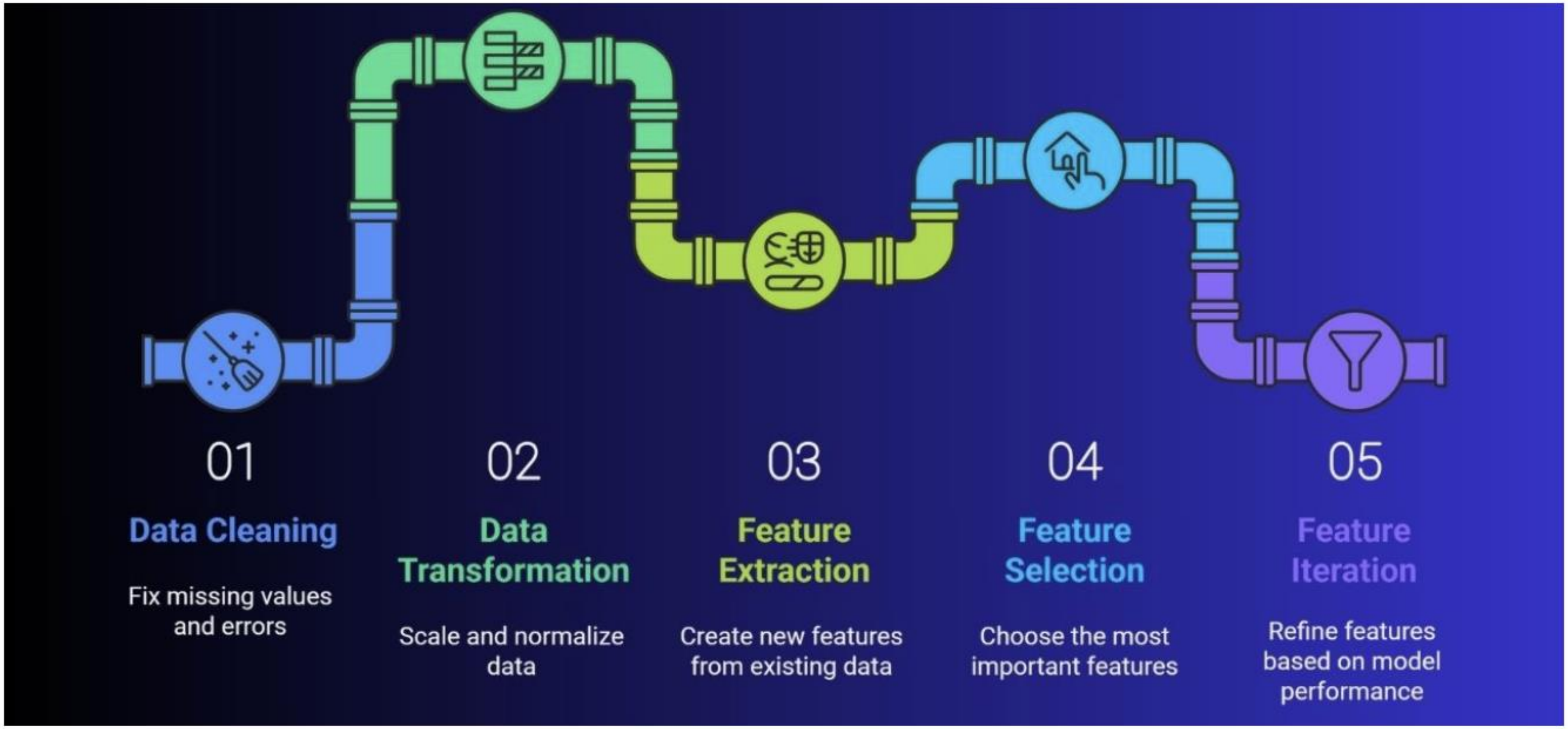
- when the periodic, oscillatory, or rhythmic structure of a time series is more informative than its raw time evolution
  - **Example:** EEG/ECG rhythms, machinery vibrations
- Frequncies have direct pysical meaning
  - **Example:** motor speed, brain rhythms(alfa-beta bands
- A long time series signal may be described by few dominant frequencies
- Classes that overlap in time domain may be well separated in frequency domain
  - **Example:** healty vs faulty motor: same mean, different vibration freq.
- Features: **dominnat frequency, spectral centroid, frequency variance etc.**
- Not good for: exact timing, non-periodic events, short signals without cycles

Time Series Features

3. Autocorrelation Based Features:

- quantify how much a time series is related to its own past values
- Measures temporal depencies
  - **Example:** EEG/ECG rhythms, machinery vibrations
- Frequncies have direct pysical meaning
  - **Example:** motor speed, brain rhythms(alfa-beta bands
- A long time series signal may be described by few dominant frequencies
- Classes that overlap in time domain may be well separated in frequency domain
  - **Example:** healty vs faulty motor: same mean, different vibration freq.
- Features: **dominnat frequency, spectral centroid, frequency variance etc.**
- Not good for: exact timing, non-periodic events, short signals without cycles

Traditional Feature Engineering



Traditional Feature Engineering

- **Data Cleaning:** Fix missing values, remove duplicates, and correct errors to ensure clean, reliable data for machine learning.
- **Data Transformation:** Scale, normalize, and encode raw data so models can easily understand and process different feature types.
- **Feature Extraction:** Create new, meaningful features by deriving or combining information from existing data to boost model learning.
- **Feature Selection:** Choose the most important features using correlation, mutual information, or model-based methods to reduce noise and overfitting.
- **Feature Iteration:** Continuously test and refine features by adding, removing, or adjusting them based on model performance improvements.

Traditional Feature Engineering Examples

Customer Churn Prediction (Telecom)

Feature engineering helps predict which customers may leave a telecom service. Useful features include monthly usage trends, call drops, complaint history, payment delays, contract type, and recent plan changes.

Fraud Detection (Banking & Payments)

Banks use feature engineering to identify suspicious transactions. Key features include transaction frequency, amount spikes, device location mismatches, login patterns, merchant risk scores, and spending behavior anomalies.

Healthcare Risk Prediction (Claims/Hospitalization)

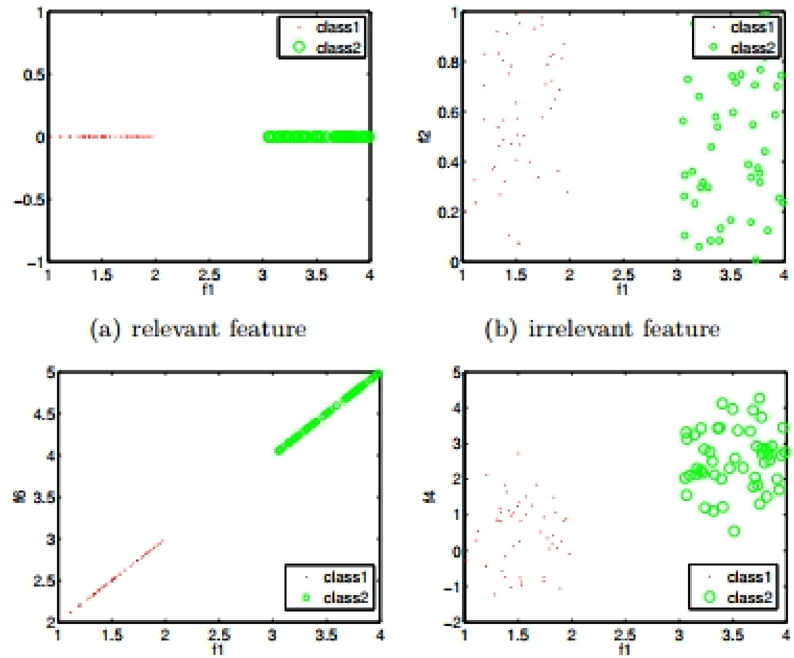
In healthcare ML models, engineered features include patient history, claim patterns, medication usage, diagnosis codes, recurring visits, vitals trends, and time since last consultation to predict hospitalization risks.

Retail & E-commerce Personalization

E-commerce platforms engineer features like browsing behavior, purchase frequency, cart activity, discount usage, product categories viewed, session duration, and seasonal trends to power personalized recommendations.

Why are not all features used?

- f1: separate two classes
- f2: can't separate
- f4: noisy feature
- f6: f1 correlated feature



(a) relevant feature (b) irrelevant feature

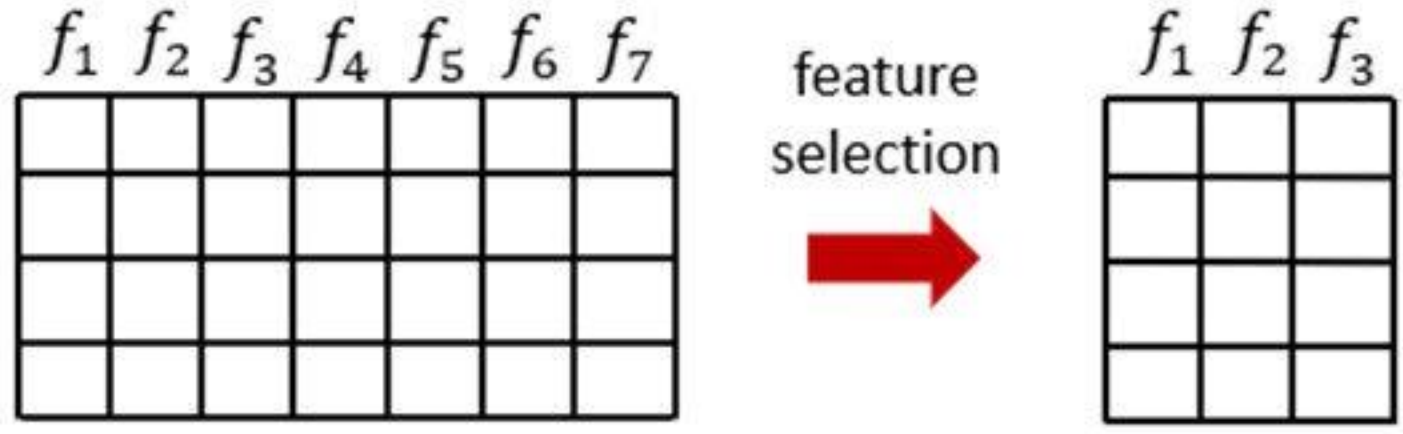
(c) redundant feature (d) noisy feature

Dimensionality Reduction

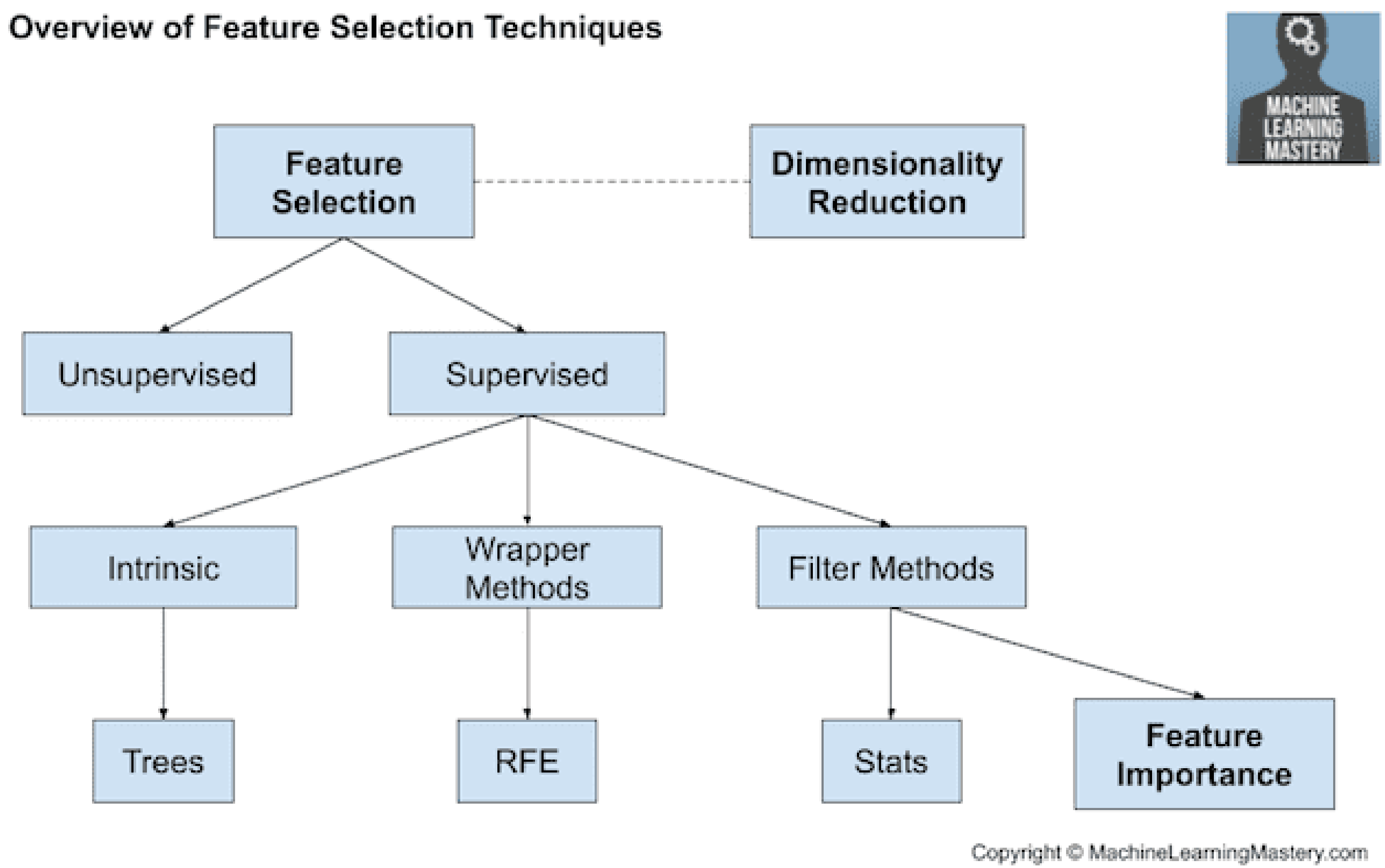
- Feature Selection: Selects characteristic features from all features

- Feature Extraction: Produce new features from all features

Feaature Selection



Feature Selection



Traditional Feature Selection Methods

- **Supervised Feature Selection**
  - Filter Based: select features **before** model training by using **statistical properties of the data**, *independent of any specific learning algorithm*.
  - Wrapper Based: evaluate subsets of features by training a machine learning model and measure its performance
  - Embedded: Özellik çıkarma öğrenme modeli tarafından yapılır.

Filter Based Feature Selection

Each feature x_j is scored using a criterion:

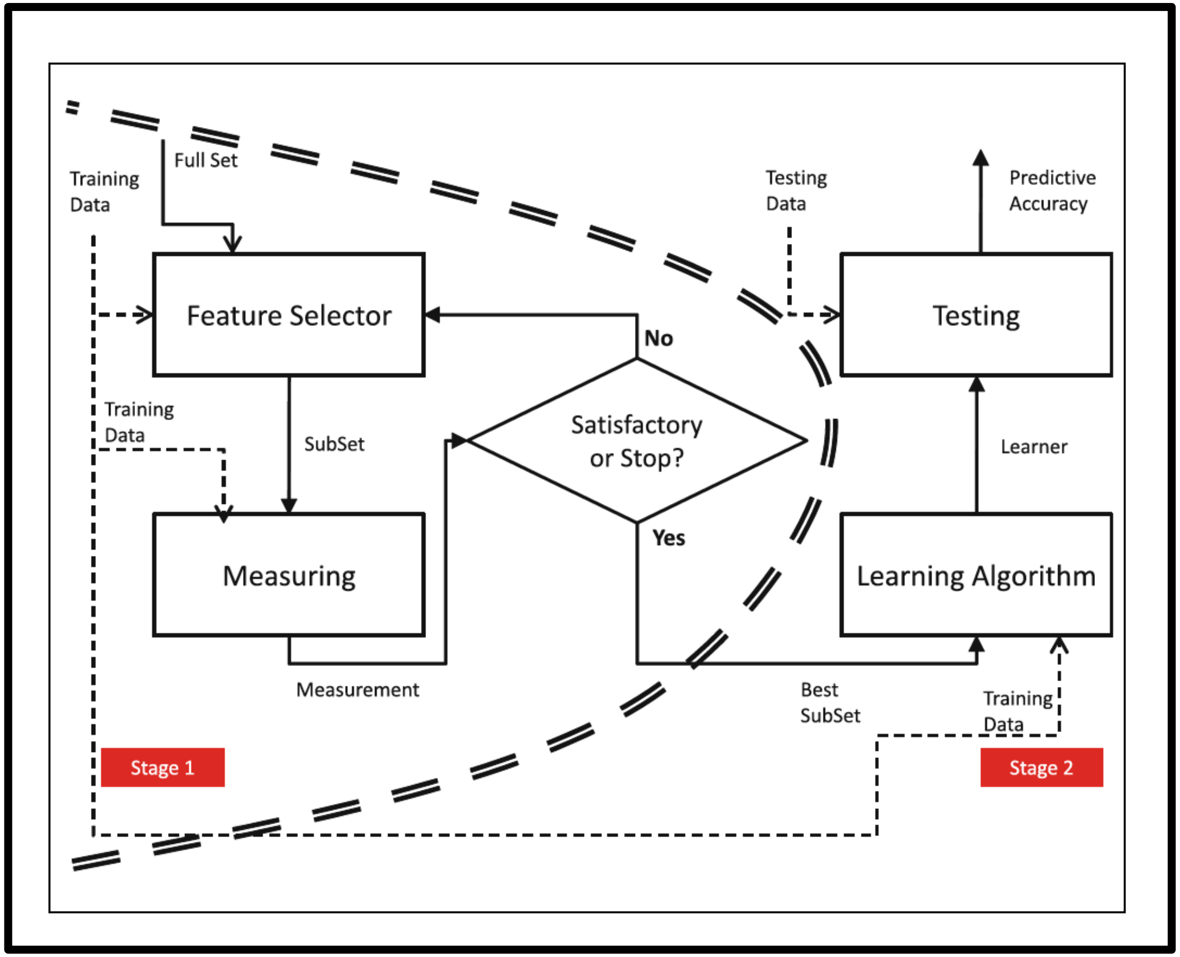
$$\text{Score}(x_j, y)$$

Then:

- Rank features
- Select top-k or those above a threshold

Filter Based Feature Selection

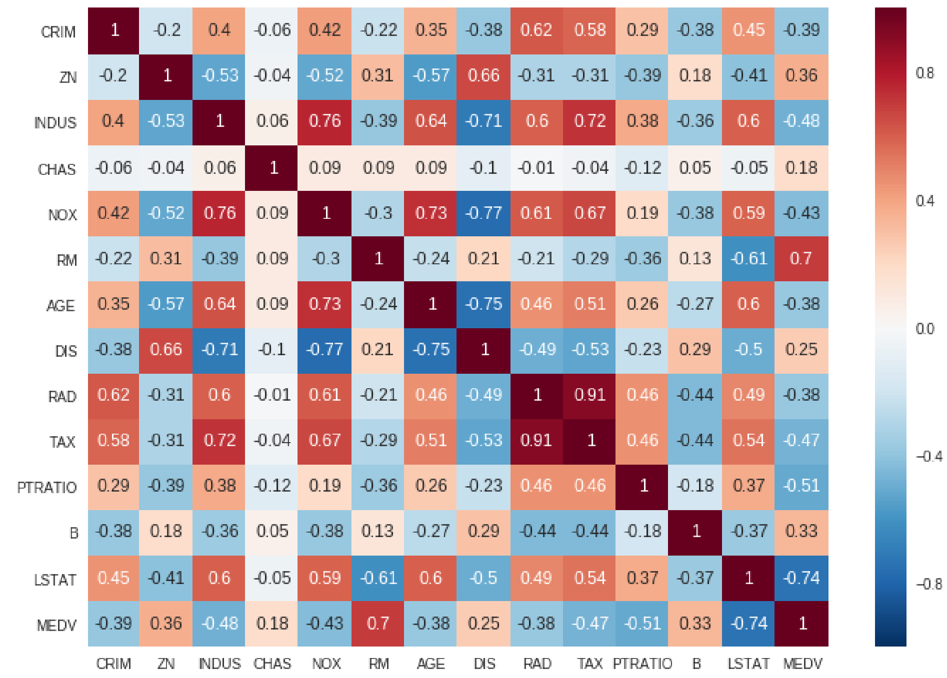
Set of all Features → **Selecting the Best Subset** → **Learning Algorithm** → **Performance**



Filter Based Feature Selection

- **Correlation based:**

$$\rho(x,y) = \frac{\mathrm{cov}(x,y)}{\sigma_x \sigma_y}$$



Variance

- **Population Variance:** Bir veri kümesindeki her örneğin **ortalamadan ne kadar uzak olduğunu gösterir.**
- **Sample Variance:** Veri kümesinin tamamı için hesap yapmak mümkün olmadığında seçilen bir grup üzerinde varyans hesaplandığı için (değer çok küçük olmasın diye (n-1)'e bölünür.

$$\text{Population Variance: } \mathrm{Var}(x) = \frac{\sum_{i=1}^{N}\left(X_i - \mu\right)^2}{N}$$

$$\text{Sample Variance: } \mathrm{Var}(x) = \frac{\sum_{i=1}^{n}\left(X_i - \bar{x}\right)^2}{n-1}$$

- **Standart Sapma:** Çoğunluğun durumunu gösterir. Örneğin bir gruptaki kişilerin ağırlık ortalaması 75 ve standart sapma 20 kg ise, insanların çoğunluğu 55 kg ile 95kg arasındadır.

Varyans ve Kovaryans

- **Varyans büyükse** örnekler birbirlerinden ve ortalamadan uzak demektir.

$$\sigma^2 = \frac{\Sigma(x - \mu)^2}{N}$$

- Kovaryans : İki değişkenin birbirleri ile ilişkisini gösterir. Negatif ve pozitif değer alabilir. Pozitif kovaryans iki değişkenin aynı yönde birlikte değiştiğini gösterir.

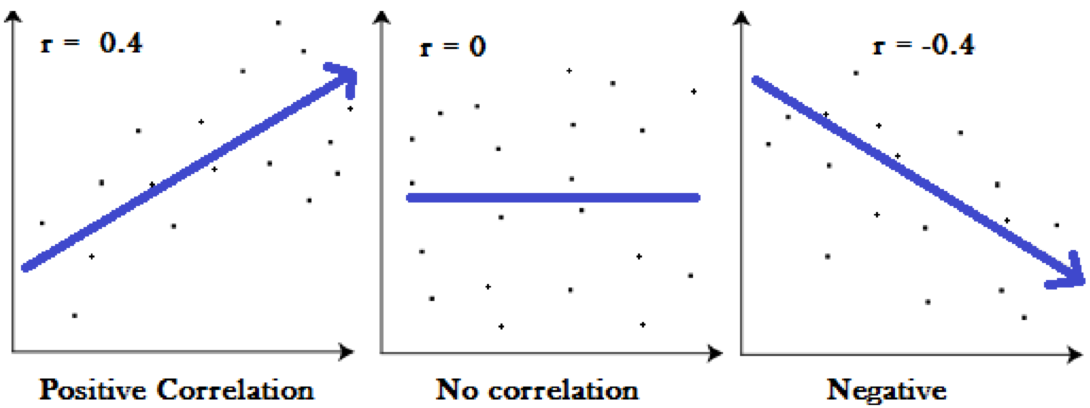
$$\text{Population Covariance: } \mathrm{Cov}(x,\ y) = \frac{\sum_{i=1}^{N}\left(X_i - \mu_x\right)\left(Y_i - \mu_Y\right)}{N}$$

$$\text{Sample Covariance: } \mathrm{Cov}(x,\ y) = \frac{\sum_{i=1}^{N}\left(X_i - \bar{x}\right)\left(Y_i - \bar{y}\right)}{n-1}$$

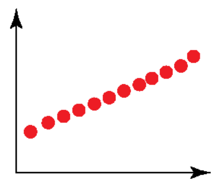
Korelasyon Nedir?

- Korelasyon iki değişken arasındaki ilişkinin kuvveti ve yönüdür. Korelasyon katsayısı bu ilişkinin [-1,+1] arasındaki ölçüm sonucudur.

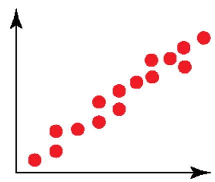
$$Correlation = \frac{Cov\ (x, y)}{\sigma x * \sigma y}$$



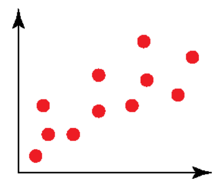
Korelasyon Nedir?



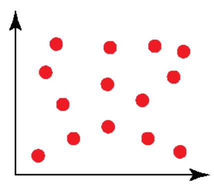
Perfect
Positive
Correlation



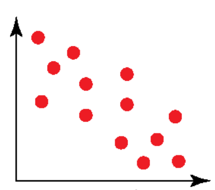
Strong
Positive
Correlation



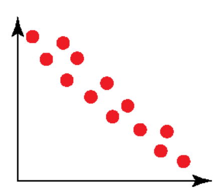
Weak
Positive
Correlation



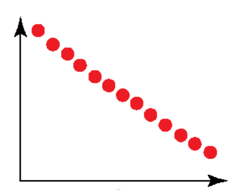
No
Correlation



Weak
Negative
Correlation



Strong
Negative
Correlation



Perfect
Negative
Correlation

Filter Based Feature Selection

- **Chi-square test:** Are two categorical variables independent?
- If a feature is useful, its values should appear unevenly across classes

$$X^2 = \frac{(Observed\ frequency - Expected\ frequency)^2}{Expected\ frequency}$$

- Observed frequency: Number of observations of class
- Expected frequency: Number of expected observations of class

Filter Based Feature Selection

- **Chi-square test:** Are two categorical variables independent?
- If a feature is useful, its values should appear unevenly across classes

$$X^2 = \frac{(Observed\ frequency - Expected\ frequency)^2}{Expected\ frequency}$$

- Observed frequency: Number of observations of class
- Expected frequency: Number of expected observations of class

Chi-square test:

Step 1️⃣ Build a contingency table

Suppose:

- Binary classification: Positive / Negative
- Feature: word = *"excellent"*

	Positive	**Negative**	**Total**
word present	40	5	45
word absent	10	45	55
Total	50	50	100

Step 2️⃣ Compute expected counts (independence assumption)

Expected value formula:

$$E_{ij} = \frac{(\text{row total}) \times (\text{column total})}{\text{grand total}}$$

Expected counts should be reasonably large

Rule of thumb: $E_{ij} \geq 5$

Example:

$$E(\text{present}, \text{positive}) = \frac{45 \times 50}{100} = 22.5$$

If feature and label were independent, we'd expect **22.5**, but we observed **40**.

Chi-square test:

Step 3️⃣ Compute Chi-square statistic

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Where:

- O = observed count
- E = expected count

📌 Large deviation → large χ^2 → **strong dependence**

Step 4️⃣ Interpretation

- **$\chi^2 \approx 0$** → independent → feature is useless
- **χ^2 large** → dependent → feature is informative

For feature selection:

- We **rank features by χ^2**
- Select **top-k** features

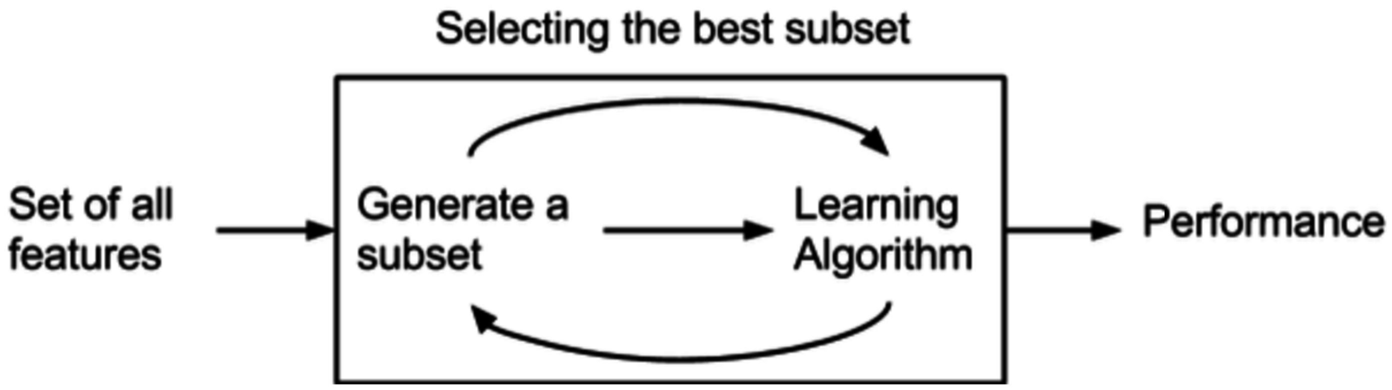
Chi-Square Olasılık Tablosu

The areas given across the top are the areas to the right of the critical value. To look up an area on the left, subtract it from one, and then look it up (ie: 0.05 on the left is 0.95 on the right)

df	0.995	0.99	0.975	0.95	0.90	0.10	0.05	0.025	0.01	0.005
1	---	---	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725	26.757
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.300
13	3.565	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688	29.819
14	4.075	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141	31.319
15	4.601	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578	32.801
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000	34.267
17	5.697	6.408	7.564	8.672	10.085	24.769	27.587	30.191	33.409	35.718
18	6.265	7.015	8.231	9.390	10.865	25.989	28.869	31.526	34.805	37.156
19	6.844	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191	38.582
20	7.434	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566	39.997
21	8.034	8.897	10.283	11.591	13.240	29.615	32.671	35.479	38.932	41.401
22	8.643	9.542	10.982	12.338	14.041	30.813	33.924	36.781	40.289	42.796
23	9.260	10.196	11.689	13.091	14.848	32.007	35.172	38.076	41.638	44.181
24	9.886	10.856	12.401	13.848	15.659	33.196	36.415	39.364	42.980	45.559
25	10.520	11.524	13.120	14.611	16.473	34.382	37.652	40.646	44.314	46.928
26	11.160	12.198	13.844	15.379	17.292	35.563	38.885	41.923	45.642	48.290
27	11.808	12.879	14.573	16.151	18.114	36.741	40.113	43.195	46.963	49.645
28	12.461	13.565	15.308	16.928	18.939	37.916	41.337	44.461	48.278	50.993
29	13.121	14.256	16.047	17.708	19.768	39.087	42.557	45.722	49.588	52.336
30	13.787	14.953	16.791	18.493	20.599	40.256	43.773	46.979	50.892	53.672
40	20.707	22.164	24.433	26.509	29.051	51.805	55.758	59.342	63.691	66.766
50	27.991	29.707	32.357	34.764	37.689	63.167	67.505	71.420	76.154	79.490
60	35.534	37.485	40.482	43.188	46.459	74.397	79.082	83.298	88.379	91.952
70	43.275	45.442	48.758	51.739	55.329	85.527	90.531	95.023	100.425	104.215
80	51.172	53.540	57.153	60.391	64.278	96.578	101.879	106.629	112.329	116.321
90	59.196	61.754	65.647	69.126	73.291	107.565	113.145	118.136	124.116	128.299
100	67.328	70.065	74.222	77.929	82.358	118.498	124.342	129.561	135.807	140.169

Wrapper Based Methods

- Trains a model repeatedly on different feature subsets and evaluating perform
- Feature usefulness is defined by **model performance**



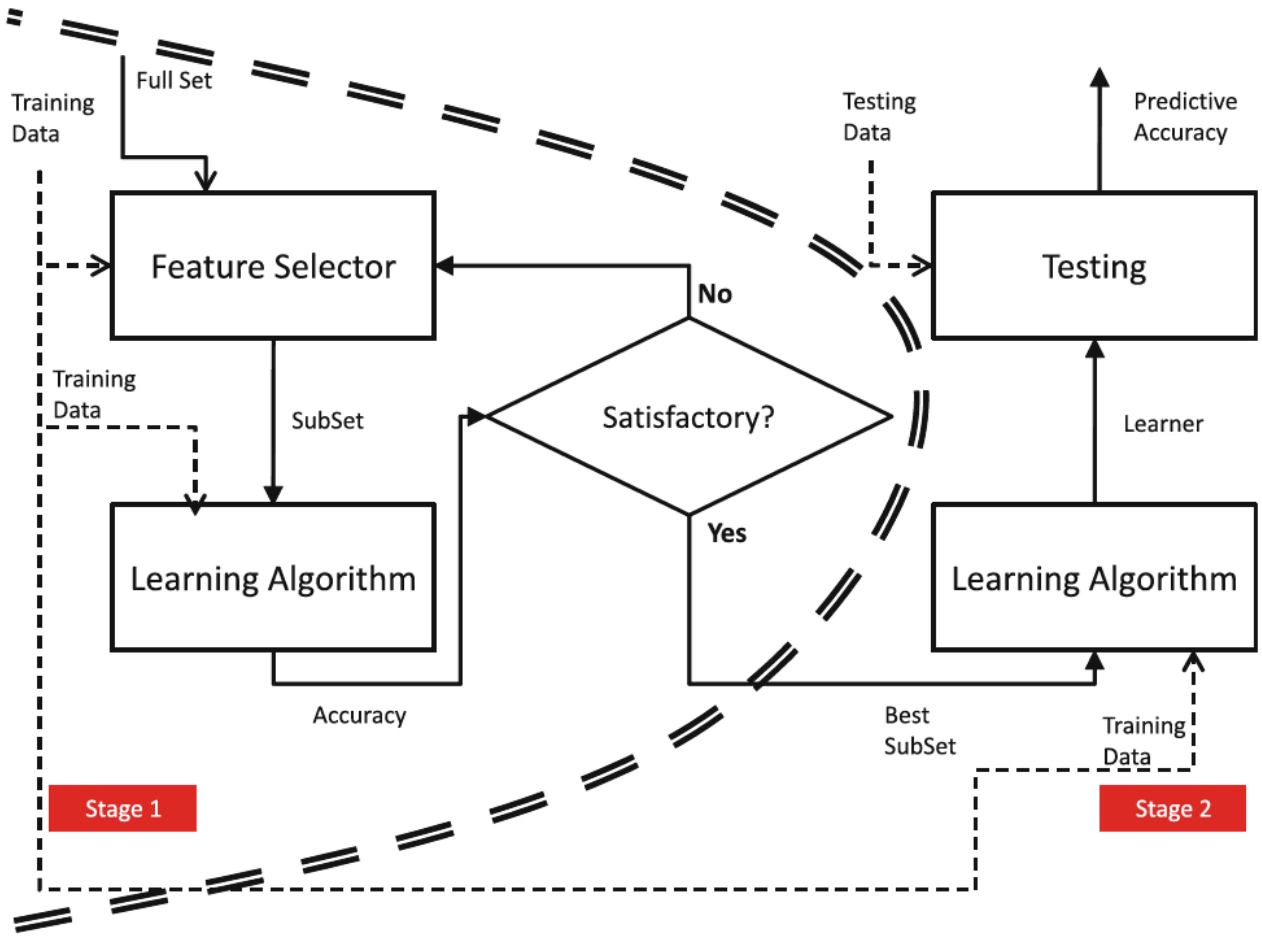
✓ Small–medium feature sets

✓ Expensive features

✓ Accuracy more important than speed

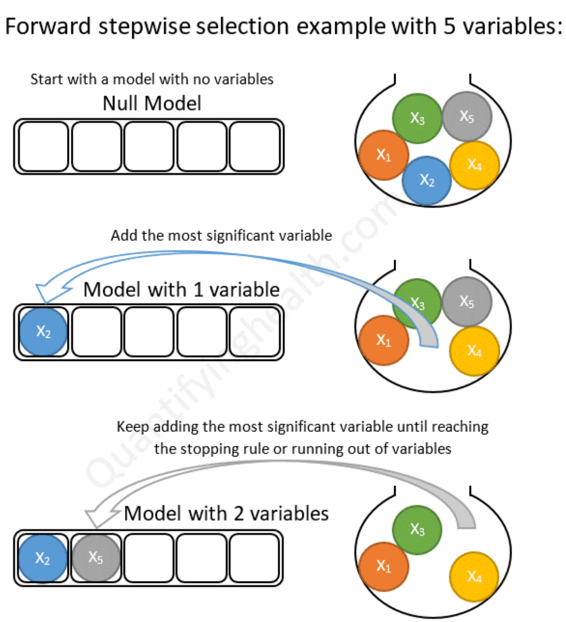
✓ Strong interactions expected

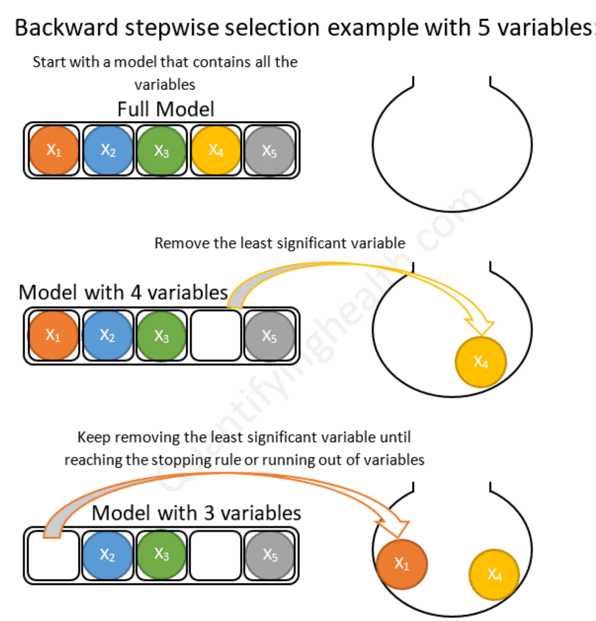
Wrapper Based Methods



Wrapper Based Methods

- Forward Selection ve backward elimination yöntemleri





Wrapper Based Methods

Steps to perform Forward Feature Selection

1. Train n model using each feature (n) individually and check the performance
2. Choose the variable which gives the best performance
3. Repeat the process and add one variable at a time
4. Variable producing the highest improvement is retained
5. Repeat the entire process until there is no significant improvement in the model's performance

Analytics Vidhya
Learn everything about analytics

Embedded Feature Selection

- Embedded methods perform feature selection during model training itself.

✓ Model-aware
✓ More efficient than wrappers
✓ Less prone to overfitting than wrappers
✓ Capture interactions (trees)

Embedded Feature Selection

- Regularization
  - LASSO (Least Absolute Shrinkage and Selection Operator (LASSO), L1
  - Ridge (L2)
  - Elastic Net (L1+L2)
- Tree Based Methods
  - Decision Tree
  - Random Forest
- Linear Models
  - SVM
  - Sparse Logistic Regression

Traditional Feature Selection Methods

Aspect	Filter	Wrapper	Embedded
Model-aware	❌	✓	✓
Speed	Fast	Slow	Medium
Feature interactions	❌	✓	✓
Generality	High	Low	Medium

Feature Extraction

- Features are transformed into a new space that captures the underlying structure of the data

Feature selection	Feature extraction
Choose subset of original features	Create new features
Keeps original meaning	Meaning is abstract
Example: remove low-variance features	Example: PCA components

Feature Extraction

Unsupervised Feature Selection Techniques

Principal Component Analysis (PCA)

PCA is a linear dimensionality reduction method that converts the original feature space into a new orthogonal space defined by principal components.

Independent Component Analysis (ICA)

ICA can be used to convert the original feature space into a new space characterized by statistically independent components.

Non-Negative Matrix Factorization (NMF)

The non-negative matrix factor (NMF) approximates a non-negative data matrix as the product of two lower-dimensional non-negative matrices.

t-distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a nonlinear dimensionality reduction method that tries to preserve the dataset's structure by reducing the difference between pairwise probability distributions in high and low-dimensional locations.

Autoencoder

An autoencoder, which is a kind of artificial neural network, learns to encode input data into a lower-dimensional representation and then decode it back to the original version.

Feature (Representation) Learning

- Representation learning is the idea of automatically **learning useful features from raw data**, instead of hand-crafting those features manually

  Raw input : X   Representation: Z= f(x)

- **Classical Pipeline:** Raw Data → **Hand-crafted Features** → Classifier → Output.
- **Representation Learning Pipeline:** Raw Data → **Learnable Transformation** → Latent Representation → Output.

Feature Engineering	Representation Learning
Manual	Automatic
Domain expertise	Data-driven
Interpretable	Often abstract
Limited scalability	Scales well
Task-specific	Often reusable

Feature (Representation) Learning

- Representation learning is the idea of automatically **learning useful features from raw data**, instead of hand-crafting those features manually

  Raw input : X Representation: Z= f(x)

- Examples:

  Text
  - Word2Vec, GloVe → word representations
  - BERT, GPT → contextual representations

  Vision
  - CNN feature maps
  - Vision Transformers

  Time series
  - Autoencoder embeddings
  - Learned temporal representations

  Audio
  - Spectrogram embeddings
  - Learned speech representations

Supervised Representation Learning

- Representations are learned from labeled data
- In supervised representation learning, representations are optimized directly for a labeled task through the loss function
- **Examples:**
  - CNN features for image classification
  - BERT learns contextual representations dynamically using self attention

Self-Supervised Representation Learning

- In Self Supervised Learning labels are automatically generated from the data itself, not manually annotated

The model creates **its own supervision signal**.

Examples:

- Word2Vec (predict context words)
- GloVe (predict co-occurrence statistics)
- BERT (masked language modeling)
- Contrastive learning

Pros:

- no labels needed
- scalable

Cons:

- representations may be generic

Unsupervised Representation Learning

Unsupervised learning focuses on data density estimation, dimensionality reduction, or clustering to extract meaningful features

Why do we need unsupervised representation learning?

- ✓ Labels are expensive
- ✓ Large unlabeled datasets exist
- ✓ Improves downstream tasks
- ✓ Enables transfer learning
- ✓ Reduces dimensionality

Unsupervised Representation Learning

Unsupervised learning focuses on data density estimation, dimensionality reduction, or clustering to extract meaningful features

Examples: Autoencoder, PCA, k-Means, contrastive learning, variational AE

Domain	Example
NLP	Word2Vec, GloVe
Vision	Autoencoders
Time series	Contrastive encoders
Audio	Self-supervised speech models