

Models (/github/ramizallahverdiyev/Models/tree/main)

/
Dimensionality_Reduction.ipynb (/github/ramizallahverdiyev/Models/tree/main/Dimensionality_Reduction.ipynb)

1: Əsas Anlayışlar

1.1 Ölçü Anlayışı (Dimension) Nədir? Feature Space

Data elmi kontekstində “ölçü” termini, adətən **xüsusiyyət (feature)** mənasını verir.

◆ Tərif:

Bir məlumat dəstindəki **hər bir asılı olmayan dəyişən** və ya **xüsusiyyət** bir ölçüdür.

◆ Nümunə:

Aşağıdakı məlumat dəstini nəzərdən keçirək:

| Çəki (kq) | Boy (sm) | Yaş |
|-----------|----------|-----|
| 70 | 175 | 25 |
| 60 | 168 | 30 |
| 80 | 180 | 22 |

- Bu datasetin ölçüsü: **3** (çünki 3 xüsusiyyət var: çəki, boy, yaş)
- Hər bir sətir: **bir müşahidə (nümunə)**
- Bu dataset: **3-ölçülü (3D) bir vektor fəzasıdır**

In []:

Müşahidə sayı: 3

Ölçü sayı (xüsusiyyət): 3

1.2 Ölçü Lənəti (Curse of Dimensionality)

Ölçü sayı artdıqca məlumatın təbiəti əhəmiyyətli dərəcədə dəyişir. Bu hadisəyə “**ölçü lənəti**” deyilir.

◆ Tərif:

Ölçü sayı artdıqca:

- Məlumatlar **seyrəkləşir (sparse)**

- Evklid məsafəsi mənasını itirir
- Modelin ümumiləşdirmə qabiliyyəti azalır
- Hesablama (cost) **artır**

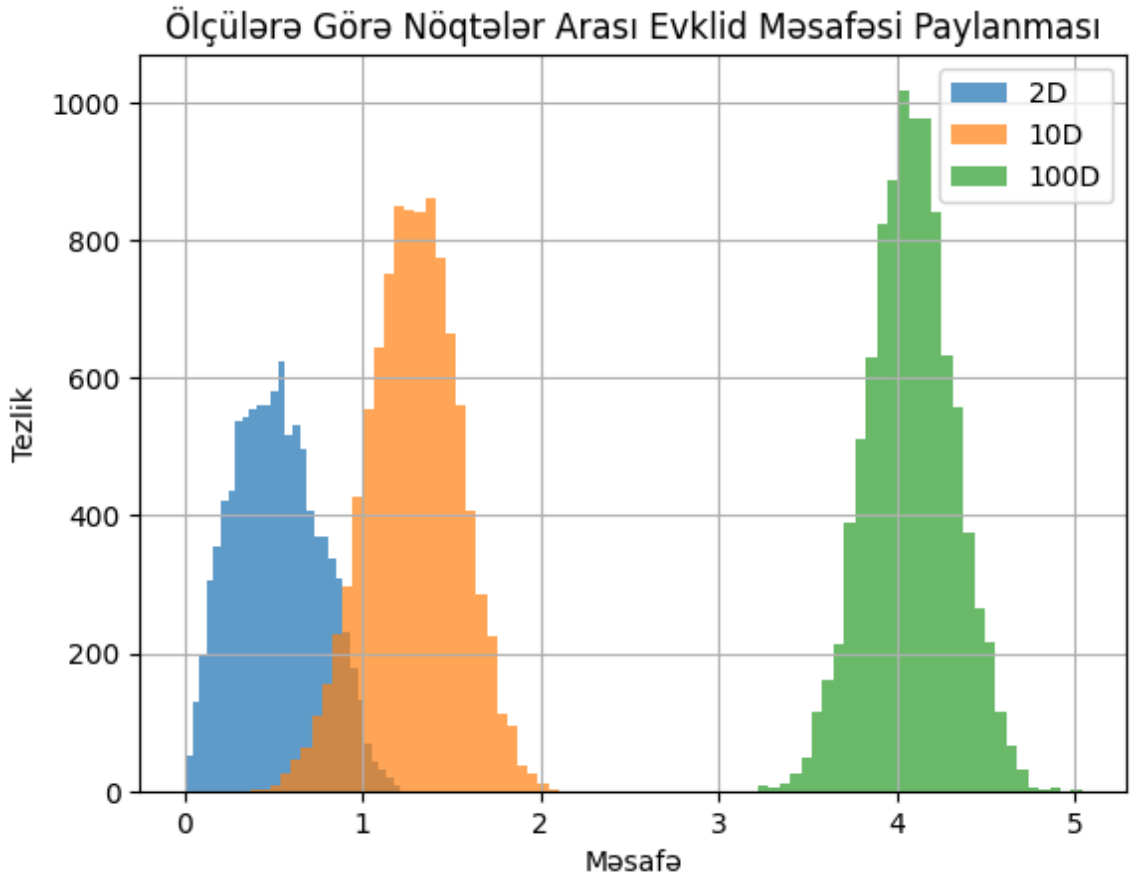
◆ Məsafənin Təsiri:

Yüksək ölçüdə, bütün nöqtələr bir-birinə oxşar məsafədə olur. Bu da məsafəyə əsaslanan alqoritmlərin (məsələn, k-NN, k-means) təsirini azaldır.

◆ Nəzəri Müşahidə:

Bir hiper-kubun içərisindəki nöqtələrin çoxu, künclərdə yerləşir. Bu, modelin məlumatlar arasında mənalı əlaqələr tapmasını çətinləşdirir.

In []:



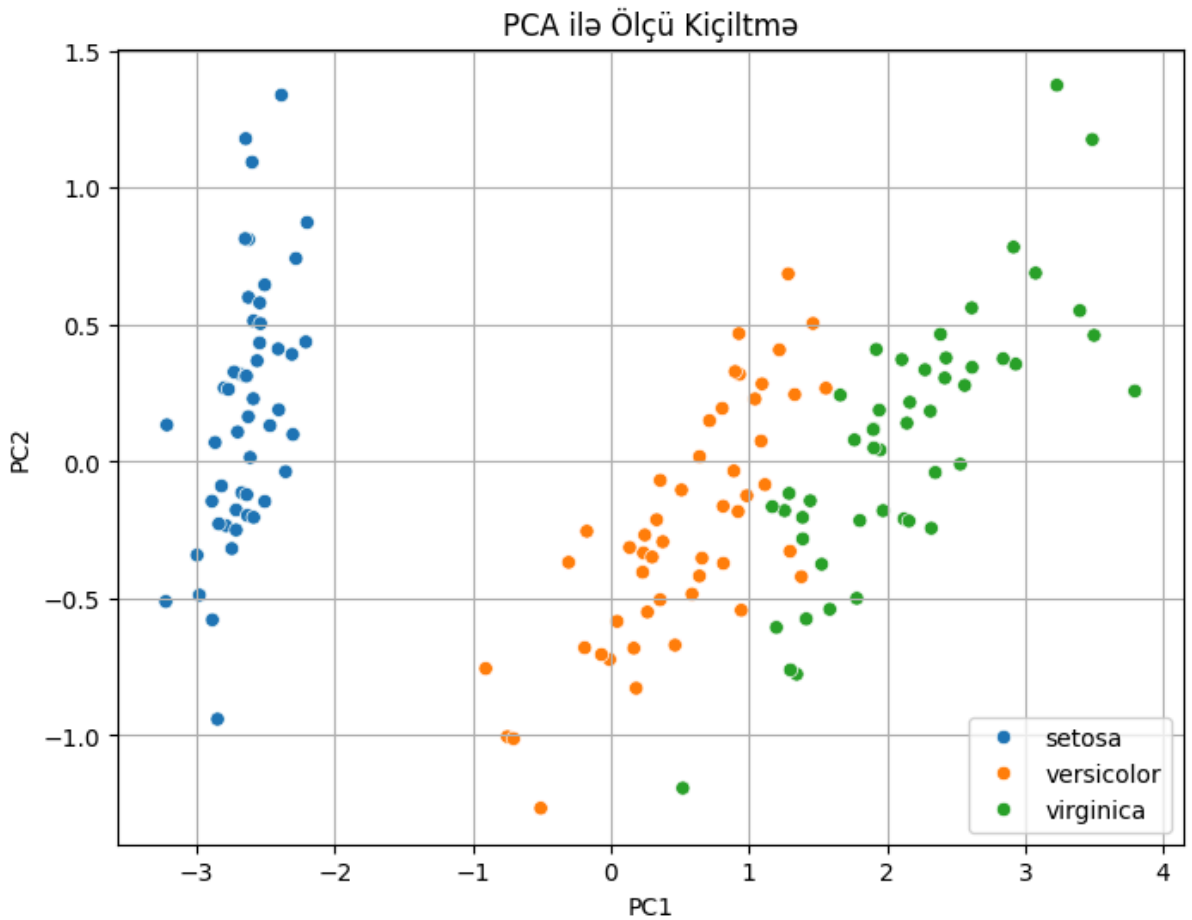


1.3 Nə üçün Ölçü Kiçiltməsi Edilir?

1. Vizualizasiya Asanlığı

→ 2D və ya 3D proyeksiyası ilə məlumat daha asan şərh olunur. Nümunə: t-SNE və ya PCA ilə məlumat nöqtələrinin qruplaşması. 2. **Yaddaş və Vaxt Səmərəliliyi** → Daha az xüsusiyyət = daha aşağı hesablama (cost) 3. **Overfitting Riskini Azaltma** → Lazımsız səs-küylü xüsusiyyətləri aradan qaldıraraq ümumiləşdirmə gücünü artırır 4. **Korelyasiyaları Ortaya Çıxarma** → PCA kimi texnikalarla bir-birinə bağlı olan xüsusiyyətlər tək komponentdə birləşdirilir 5. **Noise Azaltma** → Aşağı dispersiyaya malik komponentləri atmaq, məlumat təmizliyinə kömək edir

In []:



1.4 Ölçü Kiçiltmə Növləri

Ölçü kiçiltməsi iki əsas yolla həyata keçirilir:



1.4.1 Feature Selection

- Mövcud xüsusiyyətlər arasından **vacib olanlar seçilir**
- Heç bir yeni xüsusiyyət yaradılmır

◆ Metodlar:

- Korelyasiya analizi

- Recursive Feature Elimination (RFE)
- Lasso regressiya (L1)

◆ **Üstünlük:**

- Şərh edilə bilmə qabiliyyəti qorunur

◆ **Nümunə:**

100 xüsusiyyət → yalnız 10-u həqiqətən işə yarayır → onları seç

1.4.2 Feature Extraction

- Mövcud xüsusiyyətlərdən **yeni birləşmiş xüsusiyyətlər yaradılır**
- Məqsəd: məlumatı daha az ölçüdə təmsil etmək

◆ **Metodlar:**

- PCA
- LDA
- t-SNE
- Autoencoder (deep learning)

◆ **Nümunə:**

100 ölçülü məlumatı, %95 dispersiyanı qorunaraq 10 ölçüyə endirilir

Tətbiqi Sadə Nümunə (PCA Öncəsi Vizualizasiya)

İki xüsusiyyəti (X1, X2) olan nöqtələrin bir düz xətt üzərində qruplaşdığını düşünək. Bu halda:

- X1 və X2 əslində eyni məlumatı daşıya bilər
- PCA kimi texnikalar bu məlumatı **tək oxda sıxışdırı bilər**

Beləliklə:

- Ölçü 2 → 1 olur
- Məlumat itkisi minimum olur

In []:



Xülasə

| Mövzu | İzah |
|---------------------------------|---|
| Ölçü | Xüsusiyyət sayı (feature count) |
| Ölçü Lənəti | Ölçü artdıqca məlumat strukturunun pozulması |
| Ölçü kiçiltmə nə üçün lazımdır? | Sürət, vizualizasiya, overfittingin qarşısının alınması, məlumat çıxarımı |
| Feature Selection | Vacib xüsusiyyətlərin seçilməsi |
| Feature Extraction | Yeni birləşmiş xüsusiyyətlər yaratma |

2: Ölçü Kiçildilməsi ilə Bağlı Riyazi Əsaslar

2.1 Xətti Cəbrin Əsasları

2.1.1 Vektorlar və Matrislər

- **Vektor:** Ədədi dəyərlərdən ibarət bir sütun və ya sətir ardıcılığı.

Nümunə: $\mathbf{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

- **Matris:** Ədədlərin ikiölçülü cədvəli.

Nümunə: $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

📌 2.1.2 Matrisin Transpozu (Transpose)

- Bir matrisin sətir və sütunlarının yer dəyişməsi.
- İfadəsi: A^T
- Nümunə: $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \Rightarrow A^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$

📌 2.1.3 Matris Hasili

İki matrisin hasili, uyğun ölçülərə sahib olmaları şərti ilə edilə bilər.

Nümunə: $A_{2 \times 3} \cdot B_{3 \times 1} = C_{2 \times 1}$

📌 2.1.4 Tərs Matris (Inverse)

- Kvadrat matrisin tərsi A^{-1} kimi göstərilir.
- Xüsusiyyət: $A \cdot A^{-1} = I$ (vahid matris)

📌 2.1.5 Eigenvalue və Eigenvector

Bunlar, ölçü kiçiltməsi alqoritmlərinin təməlidir (xüsusilə PCA-da).

◆ Tərif:

Bir kvadrat matris A və $\mathbf{v} \neq 0$ vektoru üçün: $A\mathbf{v} = \lambda\mathbf{v}$

Burada:

- $\mathbf{v} \rightarrow$ eigenvector (istiqamət)
- $\lambda \rightarrow$ eigenvalue (miqyaslama miqdarı)

◆ İzah:

Matris A , vektorun yalnız uzunluğunu dəyişir, lakin istiqamətini dəyişmirsinizsə, o vektor **eigenvectordur**.

🔍 2.2 SVD (Singular Value Decomposition)

📌 Tərif:

Hər hansı bir ölçülü matris X aşağıdakı kimi parçalana bilər:

$$X = U\Sigma V^T$$

- U : sol singular vektorlar (xüsusi vektorlar)
- Σ : singular dəyərlər (diaqonal matris)
- V^T : sağ singular vektorlar (xüsusi vektorlar)

SVD Nə İş Görür?

- PCA-nın əsası SVD-dir.
- Verilərin istiqamətlərini (əsas quruluşunu) və əhəmiyyətini ortaya çıxarır.
- Noise-i azaltmaq və məlumat sıxlığını qorumaq üçün istifadə olunur.

2.3 Kovariasiya Matrisi

Ölçü kiçiltməsi metodlarının əksəriyyəti, verilərdəki xüsusiyyətlərin birlikdə necə dəyişdiyini analiz edir.

Tərif:

Datasetdəki hər bir xüsusiyyət cütü arasındakı kovariasiyanı göstərən kvadrat matrisdir.

$$\text{cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$$

Nümunə:

3 xüsusiyyətli verilər üçün kovariasiya matrisi:

$$\text{Cov} = \begin{bmatrix} \text{Var}(X_1) & \text{Cov}(X_1, X_2) & \text{Cov}(X_1, X_3) \\ \text{Cov}(X_2, X_1) & \text{Var}(X_2) & \text{Cov}(X_2, X_3) \\ \text{Cov}(X_3, X_1) & \text{Cov}(X_3, X_2) & \text{Var}(X_3) \end{bmatrix}$$

PCA və Kovariasiya:

- PCA, bu matrisi **eigenvalue və eigenvectorlara** ayıraraq, hansı oxlarda nə qədər dispersiyanın olduğunu tapır.

2.4 Məsafə Ölçüləri

Ölçü kiçiltməsində nöqtələr arasındakı uzaqlıq hesablamaları çox vacibdir.

2.4.1 Evklid Məsafəsi (Euclidean)

İki nöqtə arasındakı klassik düz xətt məsafəsi:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

2.4.2 Manhattan Məsafəsi

Yalnız üfüqi və şaquli yerdəyişmə:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i|$$

📌 2.4.3 Kosinus Oxşarlığı

Vektorlar arasındakı bucağa diqqət yetirir:

$$\cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||}$$

📌 2.4.4 Manhattan Məsafəsi

Dispersiya və kovariasiya məlumatını ehtiva edən normallaşdırılmış məsafədir.

$$D_M(\mathbf{x}, \mu) = \sqrt{(\mathbf{x} - \mu)^T S^{-1} (\mathbf{x} - \mu)}$$

- Burada S kovariasiya matrisi, μ isə orta vektordur.
- Xüsusilə çoxölçülü kənar dəyərlərin aşkarlanmasında təsirlidir.

📄 Xülasə

| Konsept | Açıqlama |
|----------------------|--|
| Matris | Ədədlərin cədvəli, verilərin təqdimatı |
| Transpoz | Sətir-sütun yer dəyişməsi |
| Xüsusi Vektor/Qiymət | İstiqaməti sabit qalan vektorlar və onların ölçüləri |
| SVD | PCA-nın əsası |
| Kovariasiya Matrisi | Xüsusiyyətlər arasındakı birlikdə dəyişikliyin əlaqəsi |
| Məsafələr | Nöqtələr arasındakı oxşarlığı ölçmə |

📚 3: Xətti Ölçü Kiçiltməsi Metodları

🗨️ Principal Component Analysis (PCA)

PCA, xətti ölçü kiçiltməsi metodları arasında ən geniş yayılmış və əsas metoddur. PCA-nın əsas məqsədi, çoxölçülü verilənlərin ən yüksək dispersiyaya malik istiqamətlərini tapmaq və veriləri bu istiqamətlər üzərinə proyeksiyalayaraq ölçüsünü azaltmaqdır. Bu istiqamətlərə **principal components** deyilir və hər biri, orijinal dəyişənlərin xətti kombinasiyasıdır.

PCA aşağıdakı addımlarla tətbiq olunur:

- Data mərkəzləşdirilir (hər sütunun ortalaması çıxarılır).
- Kovariasiya matrisi** hesablanır.
- Kovariasiya matrisinin eigenvalues və eigenvectors-u tapılır.
- Ən böyük xüsusi qiymətlərə sahib komponentlər seçilərək data bu komponentlərə proyeksiyalanır.

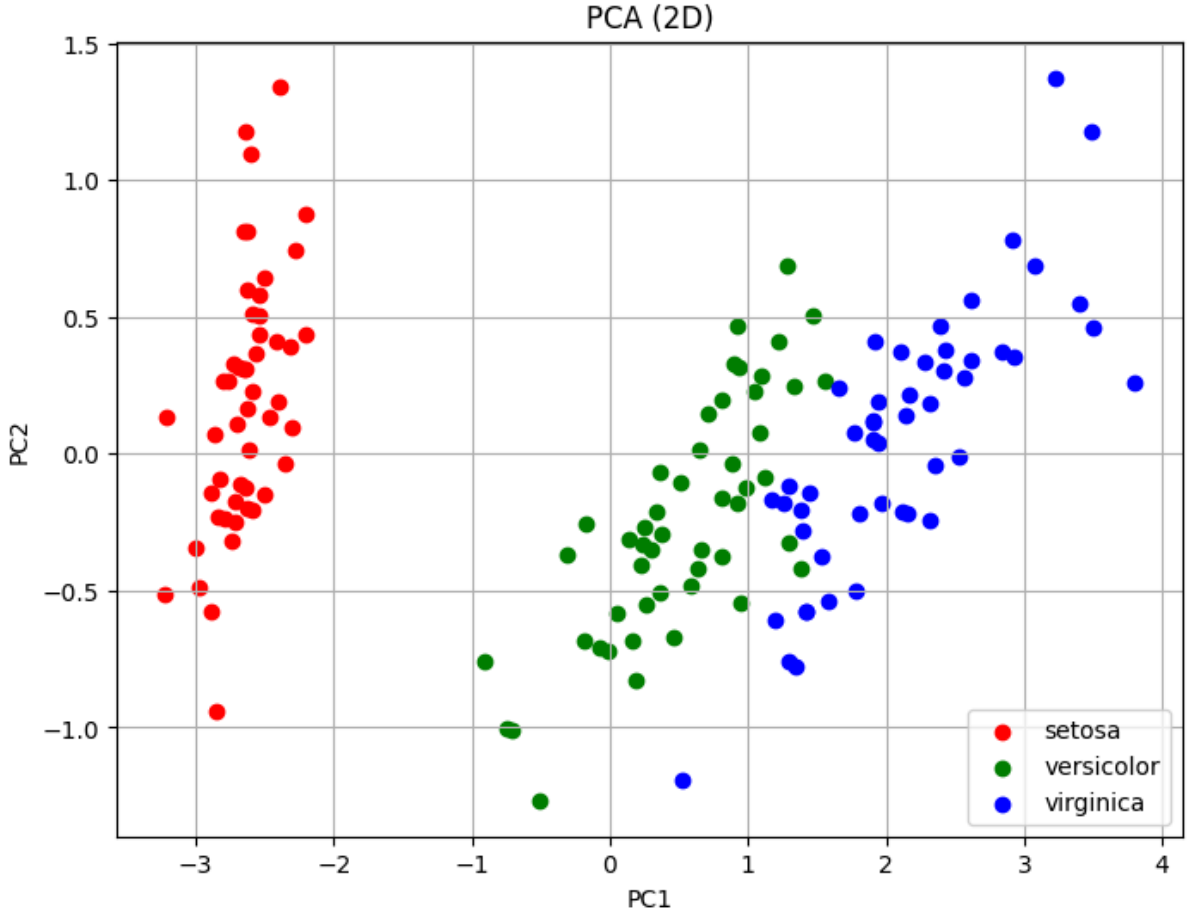
Riyazi ifadəsi: $Z = XW$

Burada:

- X : Mərkəzləşdirilmiş data matrisi (n nümunə, d xüsusiyyət)
- W : İlk k eigenvectoru ehtiva edən matris (ölçü: $d \times k$)
- Z : Ölçüsü azaldılmış yeni dataset (ölçü: $n \times k$)

✦ **Qeyd:** PCA, label tələb etməyən **unsupervised** bir metoddur. Ən çox vizualizasiya, noise azaltma və feature engineering kimi sahələrdə istifadə olunur. Lakin yalnız **xətti əlaqələri** tuta bilər və əldə edilən komponentlərin şərhə asan olmaya bilər.

In []:



🧠 Linear Discriminant Analysis (LDA)

LDA, PCA-dan fərqli olaraq **supervised** bir ölçü kiçiltməsi metodudur. LDA'nın əsas hədəfi, fərqli sinifləri bir-birindən ən yaxşı ayıran xətti oxları tapmaqdır. Yəni:

- **Sinif daxili fərqləri** minimuma endirir.
- **Siniflər arası fərqləri** maksimuma çatdırır.

Riyazi olaraq:

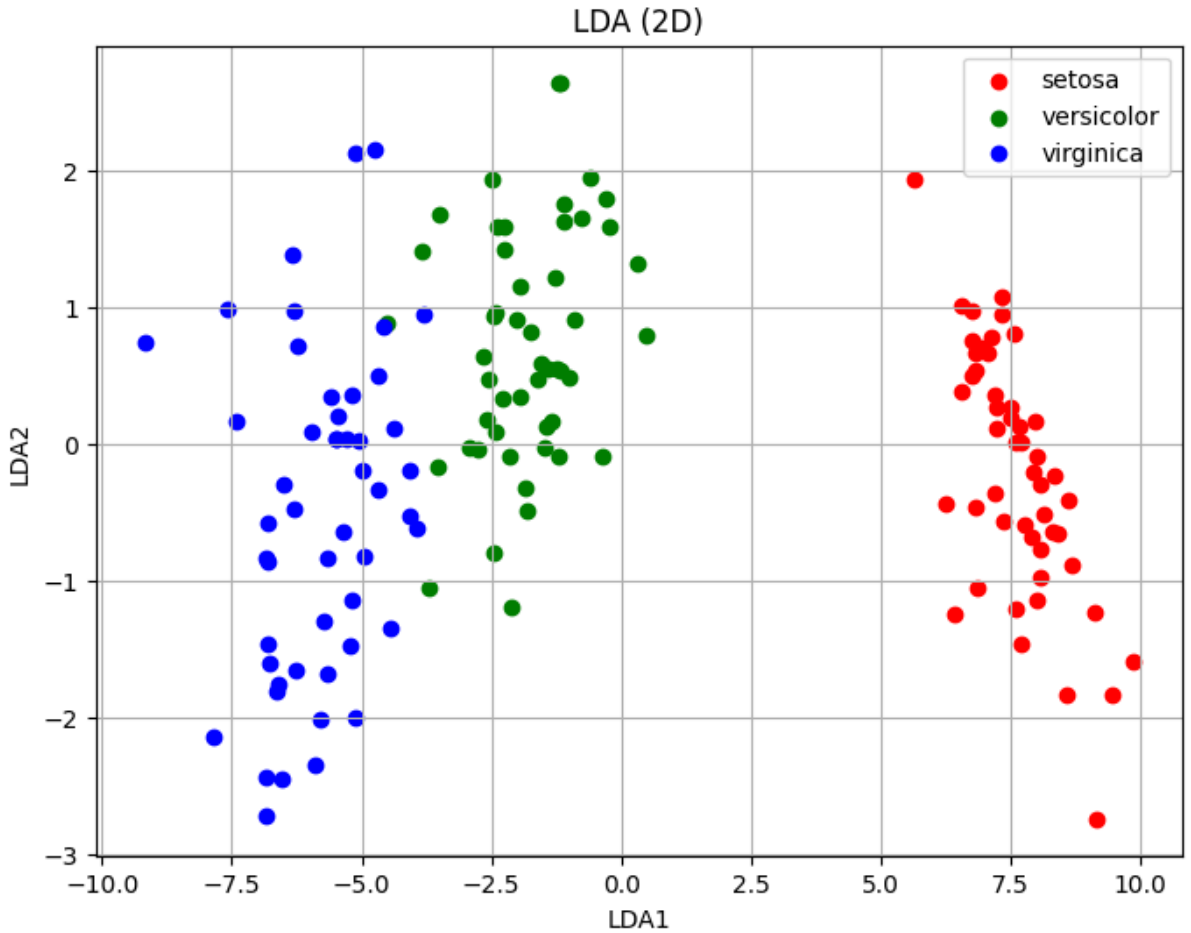
$$W = \arg \max \left(\frac{|W^T S_B W|}{|W^T S_W W|} \right)$$

Burada:

- S_W : **Sinif daxili scatter matrisi** (within-class scatter)
- S_B : **Siniflər arası scatter matrisi** (between-class scatter)

✦ LDA, ən çox $C - 1$ ölçülü bir fəzaya kiçiltmə edə bilər (C : sinif sayı). Bu metod, xüsusilə təsnifat (classification) problemlərində güclüdür; lakin sinif paylanması qeyri-bərabərdirsə və ya verilərdə siniflər xətti olaraq ayrılmırsa təsirli olmaya bilər.

In []:



🧠 Independent Component Analysis (ICA)

ICA, PCA ilə bəzi oxşarlıqlara sahib olsa da fərqli bir hədəfə malikdir. ICA, müşahidə olunan siqnalların bir-birindən **statistik olaraq müstəqil** gizli mənbələrin qarışığı olduğunu fərz edir.

Model: $X = AS$

Burada:

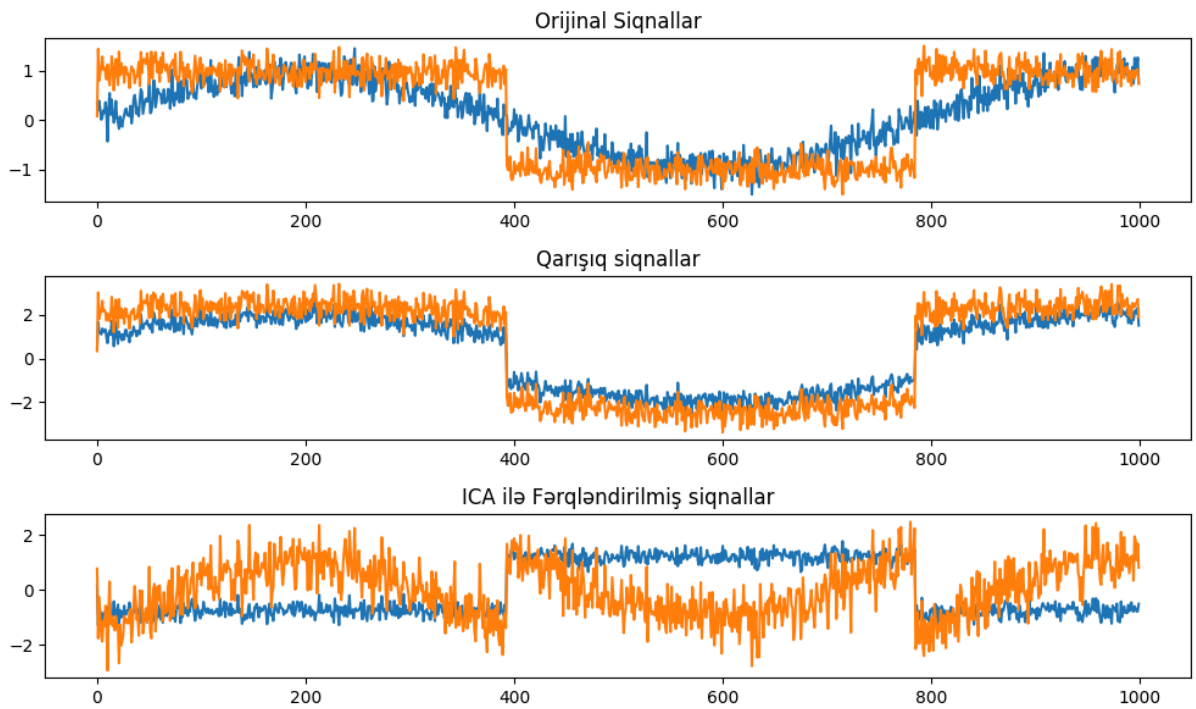
- X : Müşahidə olunan data (qarışıq siqnallar)
- A : Qarışdırıcı matris
- S : Müstəqil komponentlər (mənbə siqnalları)

Məqsəd: $S = A^{-1}X$

Nümunə: **Kokteyl Partisi Problemi** — Bir otaqda birdən çox insan danışdığında, ICA sayəsində qarışıq səslər fərqləndirilə bilər.

✦ ICA, PCA-dan fərqli olaraq **yüksək dərəcədə statistik müstəqillik** fərz edir və siqnal/şəkil fərqləndirmə kimi sahələrdə təsirlidir. Lakin hesablama baxımından daha mürəkkəbdir.

In []:



Factor Analysis (FA)

FA, müşahidə olunan dəyişənlər arasındakı korrelyasiyaları izah etmək üçün daha az sayda **gizli faktor** müəyyən edir.

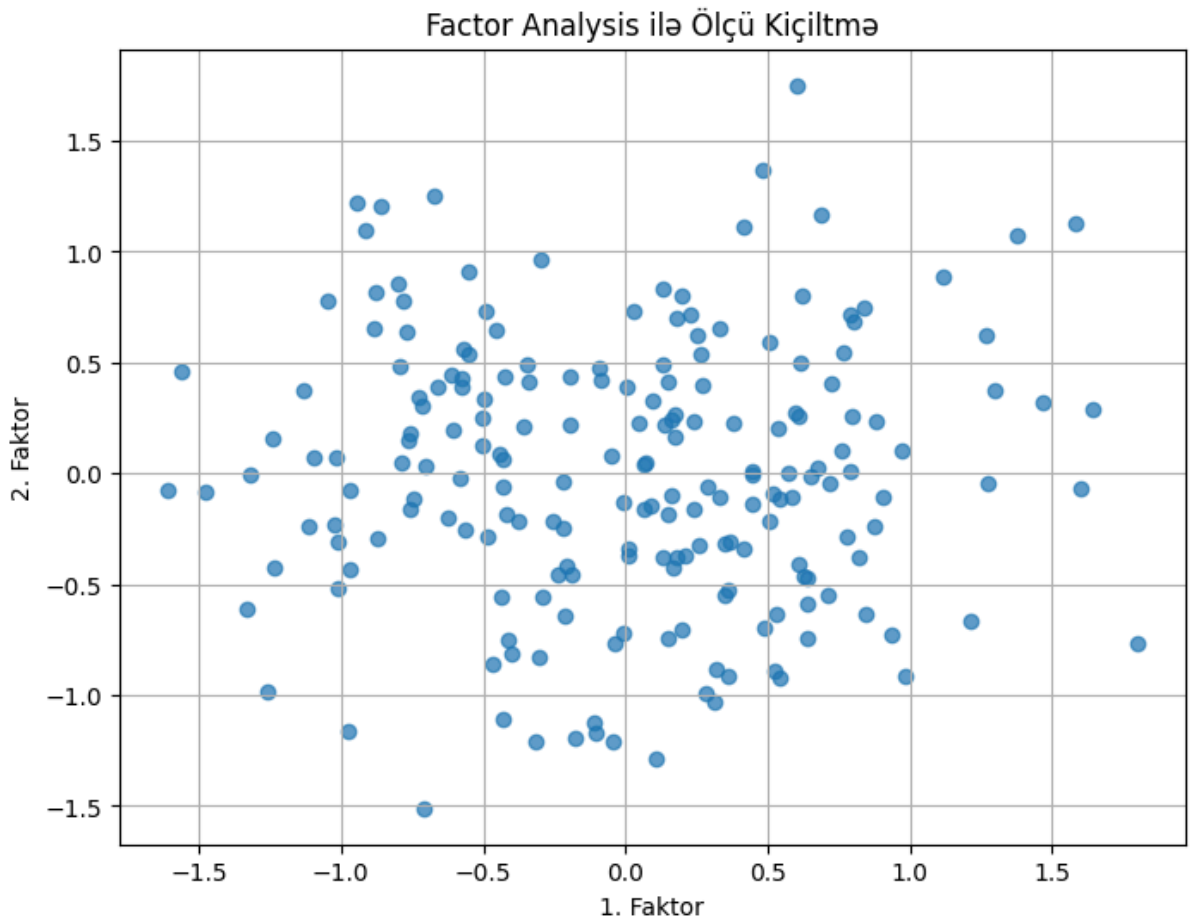
$$\text{Model: } X = \mu + LF + \epsilon$$

Burada:

- X : Müşahidə olunan data vektoru
- μ : Ortalama
- L : Yükləmə matrisi (faktorların dəyişən üzərindəki təsiri)
- F : Gizli faktorlar
- ϵ : Təsadüfi səhv həddi

✦ FA, xüsusilə sosial elmlərdə və psixologiyada geniş yayılmışdır. Səhv həddini modelə daxil etməsi, müşahidə səhvi ehtiva edən verilərdə üstünlük təmin edir. Lakin hesablama olaraq daha mürəkkəb ola bilər.

In []:



Metodların Müqayisəsi

| Metod | Növü | Hədəf | Etiket Tələb Edilir mi? | Ən Uyğun Sahələr |
|-------|--------------|---|-------------------------|--------------------------------------|
| PCA | Unsupervised | Dispersiyanı maksimuma çatdıran komponentlər tapmaq | ✗ | Vizualizasiya, noise azaltma |
| LDA | Supervised | Sınıfları ən yaxşı ayıran vektorları tapmaq | ✓ | Təsnifat |
| ICA | Unsupervised | Müstəqil mənbə komponentləri fərqləndirmək | ✗ | Siqnal emalı, səs/şəkil fərqləndirmə |
| FA | Unsupervised | Gizli faktorları kəşf etmək | ✗ | Sosial elmlər, psixometriya |

4: Xətti Olmayan Ölçü Kiçiltməsi Metodları

4.1. Manifold

Manifold nədir?

Riyaziyyatda **manifold**, hər nöqtəsi yerli olaraq Evklid fəzasına bənzəyən, lakin ümumi quruluşu daha mürəkkəb ola bilən səthlərdir. Məsələn:

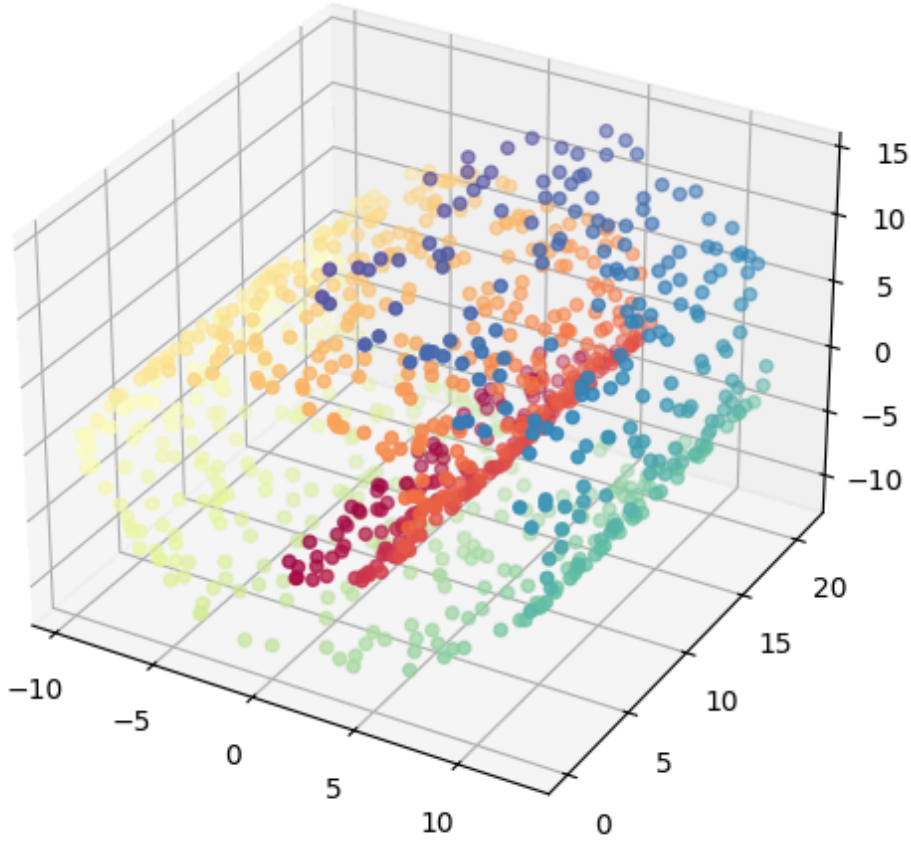
- Dünya səthi **kürə** şəklindədir (2D manifold), ancaq 3D fəzada yer alır.
- Bir dataset 100 ölçülü ola bilər, amma əslində 2 və ya 3 ölçülü bir manifold üzərində yerləşə bilər.

Yüksək ölçülü verilənlərin aşağı ölçülü səthlərdə olması

Bir çox real dünya dataseti, yüksək ölçülərdə olmasına baxmayaraq, **aşağı ölçülü manifold** üzərində paylanmışdır. Manifold öyrənmə texnikaları, bu aşağı ölçülü quruluşu ortaya çıxarmağı hədəfləyir.

In []:

Swiss Roll: (3D)



4.2. 🧠 t-SNE (t-Distributed Stochastic Neighbor Embedding)

t-SNE necə işləyir?

t-SNE, data nöqtələri arasındakı **oxşarlıqları** həm yüksək, həm də aşağı ölçülü fəzalarda ehtimal paylanmaları ilə təmsil edir.

1. Yüksək ölçülü fəzada, iki nöqtə arasındakı oxşarlıq:

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$

2. Aşağı ölçülü fəzada oxşarlıq, Student-t paylanması ilə hesablanır:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$

3. Məqsəd, bu iki paylanma arasındakı **Kullback-Leibler (KL) fərqi**ni minimuma endirməkdir.

$$KL(P \parallel Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Hiperparametrlər:

- **perplexity**: Nöqtələr arası qonşuluq miqyasını müəyyən edir.
- **learning rate**: Optimizasiyada addım ölçüsü.

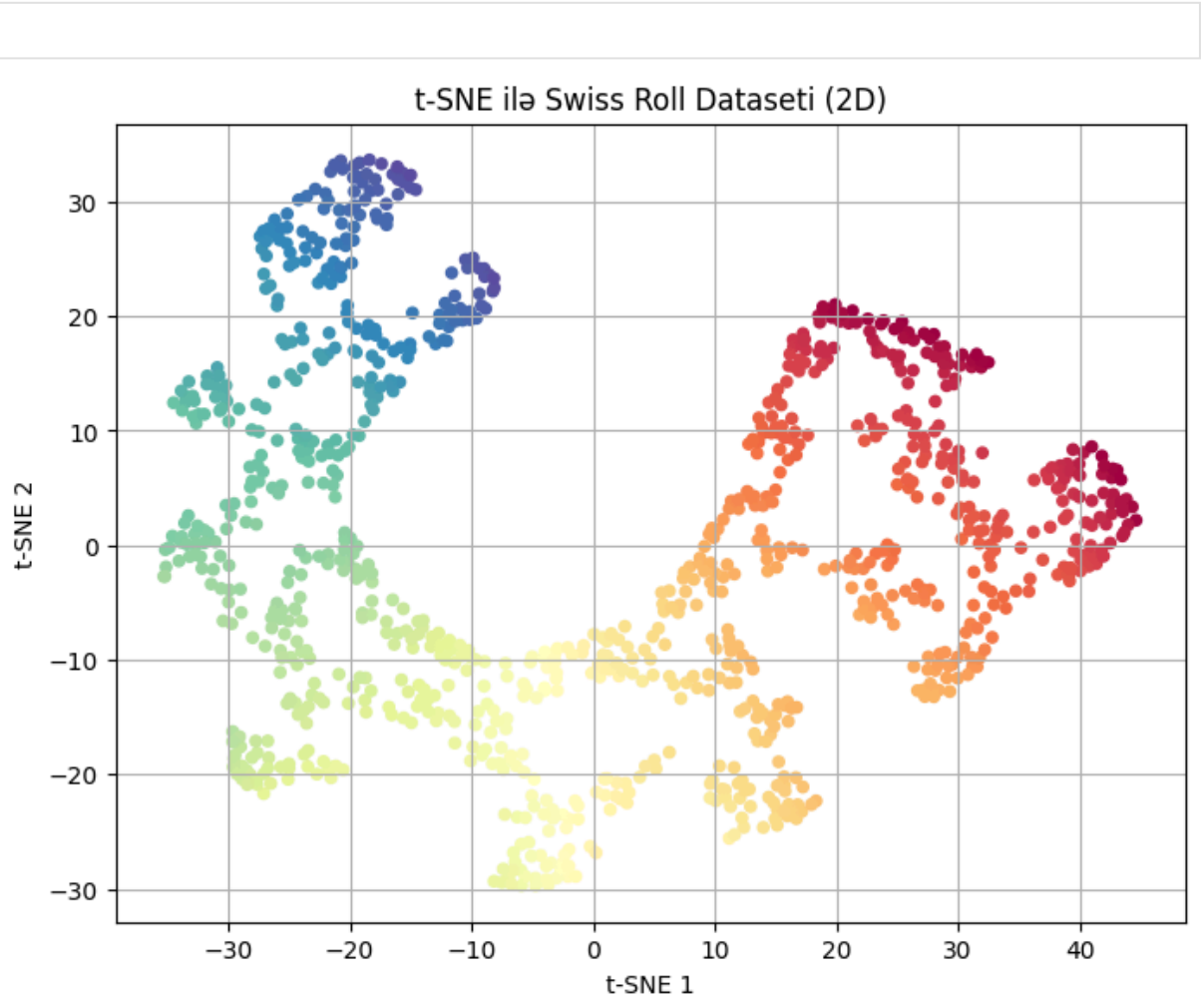
Üstünlükləri:

- Mürəkkəb datasetlərdəki qrupları vizualizasiyada mükəmməl.
- Xüsusilə 2D və 3D vizualizasiyalarda təsirli.

Çatışmazlıqları:

- Cost yüksəkdir.
- Ölçü kiçiltmə nəticəsində metrik miqyas qorunmur.

In []:



4.3. 🧠 UMAP (Uniform Manifold Approximation and Projection)

UMAP necə işləyir?

UMAP, manifold öyrənməni **qraf nəzəriyyəsi** əsaslı bir yanaşma ilə edir:

- Yüksək ölçülü fəzada nöqtələr arası qonşuluq qrafı qurulur.
- Bu quruluş aşağı ölçülü fəzada qorunacaq şəkildə optimizasiya edilir.

UMAP və t-SNE müqayisəsi

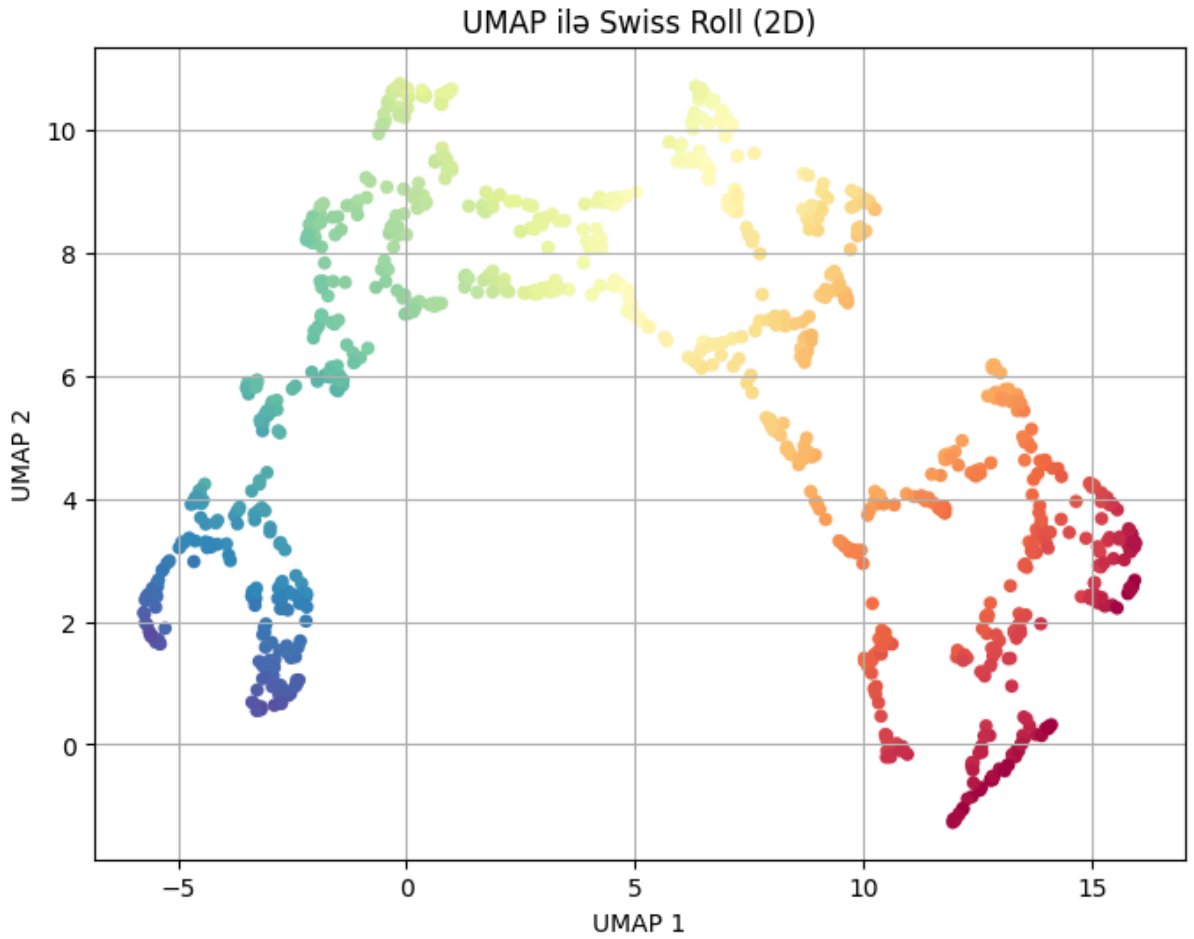
| Xüsusiyyət | t-SNE | UMAP |
|----------------|------------|-------------------------------------|
| Sürət | Daha yavaş | Daha sürətli |
| Miqyaslanma | Orta | Çox yaxşı |
| Qrup ayrışması | Güclü | Güclü + Qrup daxili quruluş qorunur |
| Parametr sayı | Az | Daha çox |

Üstünlükləri:

- Çox daha sürətli işləyir.
- Qrup quruluşunu və qrup daxili məsafələri qorumada uğurlu.
- Böyük datasetlərdə təsirli.

In []:

```
/usr/local/lib/python3.11/dist-packages/umap/umap_.py:1952: UserWarning: n_jobs value 1 overridden to 1 by setting random_state. Use no seed for parallelism.
  warn(
```



4.4. 🧠 Isomap

Geodezik məsafə konsepti

Isomap, data nöqtələri arasındakı **geodeik məsafəni** (manifold üzərindəki ən qısa yol) istifadə edir.

- Evklid məsafəsi yerinə, manifold boyunca qonşular üzərindən ən qısa yol hesablanır.

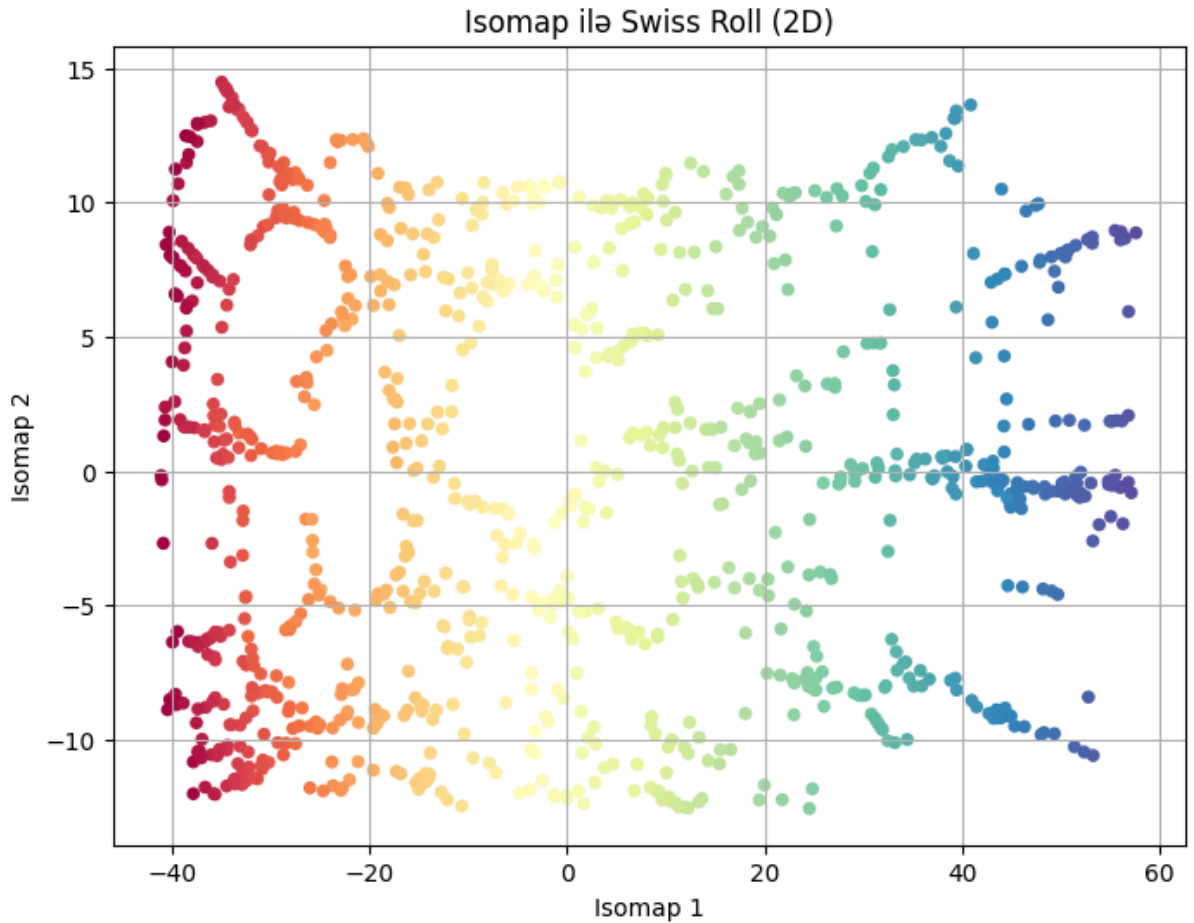
Addımlar:

1. Qonşuluq qrafı yaradılır (k ən yaxın qonşu və ya məsafə həddi).
2. Qraf üzərində bütün nöqtələr arası **geodezik məsafələr** hesablanır.
3. Bu məsafələr üzərindən **MDS (Çoxölçülü miqyaslama)** tətbiq edilərək aşağı ölçülü təmsil tapılır.

✦ Üstünlük: Mürəkkəb quruluşları tuta bilər.

✦ Çatışmazlıq: Noise-ə həssasdır.

In []:



4.5. 🧠 Locally Linear Embedding (LLE)

Əsas fikir:

Hər data nöqtəsi, qonşuları ilə **xətti kombinasiya** olaraq ifadə edilə bilər. LLE, bu yerli xətti əlaqələri aşağı ölçülü fəzada qorumağa çalışır.

Addımlar:

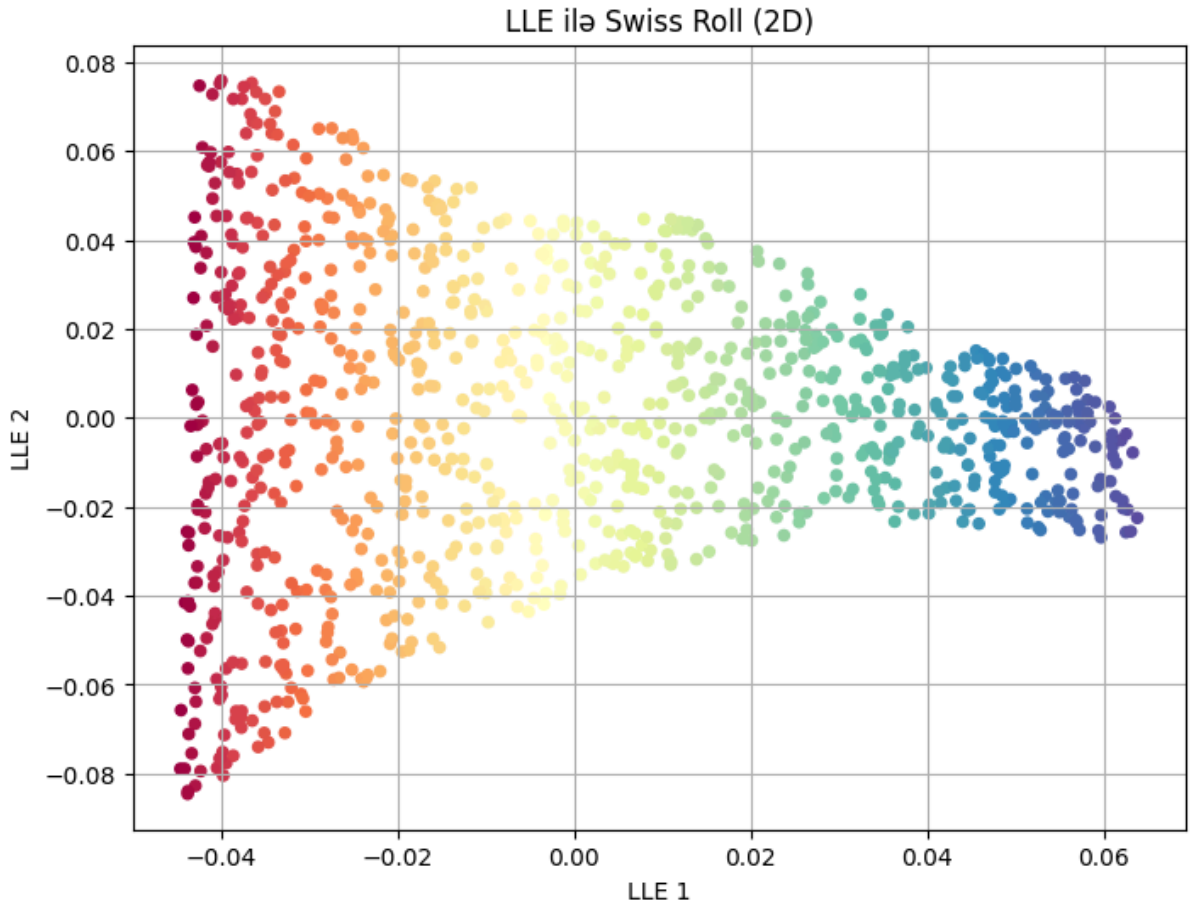
1. Hər nöqtə üçün ən yaxın qonşular müəyyən edilir.
2. Aşağıdakı çəkilər tapılır:

$$\sum_j w_{ij} x_j \approx x_i$$

Burada w_{ij} qonşuluq çəkiləridir. 3. Aşağı ölçülü fəzada bu çəki əlaqələrini qoruyacaq koordinatlar hesablanır.

- ✦ Üstünlük: Yerli quruluşu yaxşı qoruyur.
- ✦ Çətinlik: Noise və qonşu sayı seçiminə həssas.

In []:



5: Feature Selection Metodları

Feature Selection metodları, adətən üç əsas kateqoriyada araşdırılır:

- **Filter Metodları** (Datadan müstəqil, preprocessing)
- **Wrapper Metodları** (Modelə əsaslanan, feedback ilə)
- **Embedded Metodlar** (Model daxilində təbii olaraq yerləşən)

5.1. Filter Metodları

Korrelyasiya

Xüsusiyyətlər ilə hədəf dəyişəni arasındakı **xətti əlaqəni** ölçür.

- Pearson korrelyasiya əmsalı ən çox istifadə olunur.
- Korrelyasiya əmsalı +1 və ya -1-ə yaxın olan xüsusiyyətlər hədəfi güclü təsir edir.
- 0-a yaxın olanlar isə zəif əlaqəyə işarə edir.

Düstur: İki dəyişən X və Y arasındakı Pearson korrelyasiyası:

$$r_{XY} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

Qeyd: Koerelyasiya yalnız xətti əlaqələri yaxalayır, mürəkkəb əlaqələrdə yanıltıcı ola bilər.

Qarşılıqlı İnformasiya (Mutual Information - MI)

İki dəyişən arasındakı **hər hansı bir əlaqə növünü** ölçür (xətti və ya xətti olmayan).

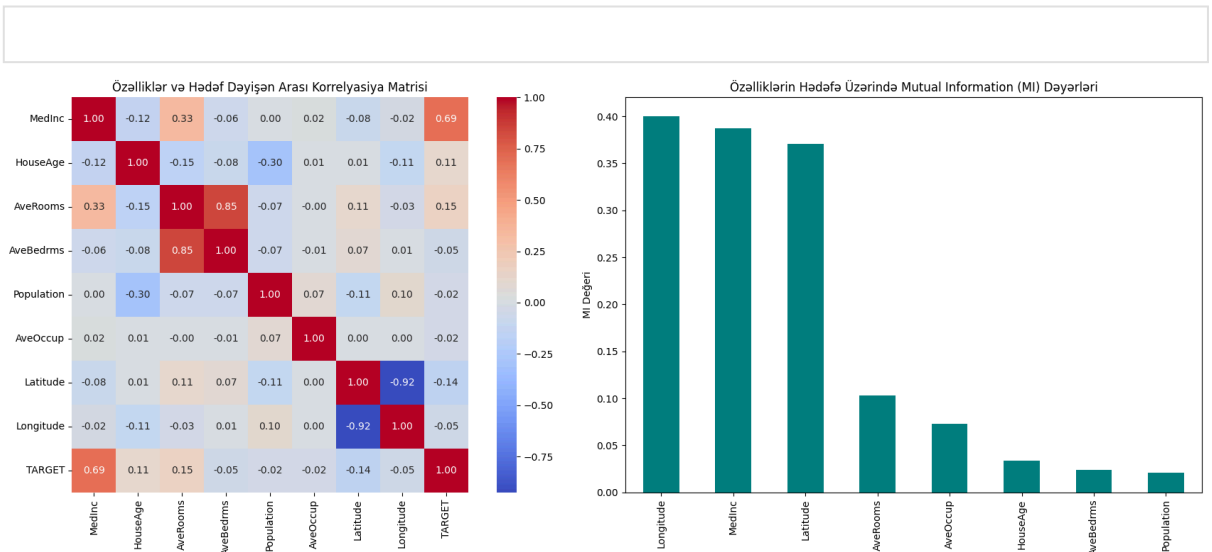
- MI, informasiya nəzəriyyəsindən gəlir və dəyişənlər arasındakı ortaq informasiyanı hesablayır.
- Yüksək MI, yüksək asılılıq anlamına gəlir.

Düstur:

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

Burada $p(x, y)$ ortaq ehtimal paylanması, $p(x)$ və $p(y)$ isə marjinal ehtimal paylanmalarıdır.

In []:



5.2. Wrapper Metodları

Recursive Feature Elimination - RFE

- Modelə əsaslanan bir metoddur.
- Başlanğıcda bütün xüsusiyyətlər istifadə olunur.
- Model öyrədilir və ən az əhəmiyyətli xüsusiyyətlər kənarlaşdırılır.
- Bu proses, müəyyən edilmiş hədəf sayına çatana qədər təkrarlanır.

✳ Məsələn, RFE ilə bir SVM və ya reqressiya modeli istifadə edilərək addım-addım az sayda ən yaxşı xüsusiyyət seçilir.

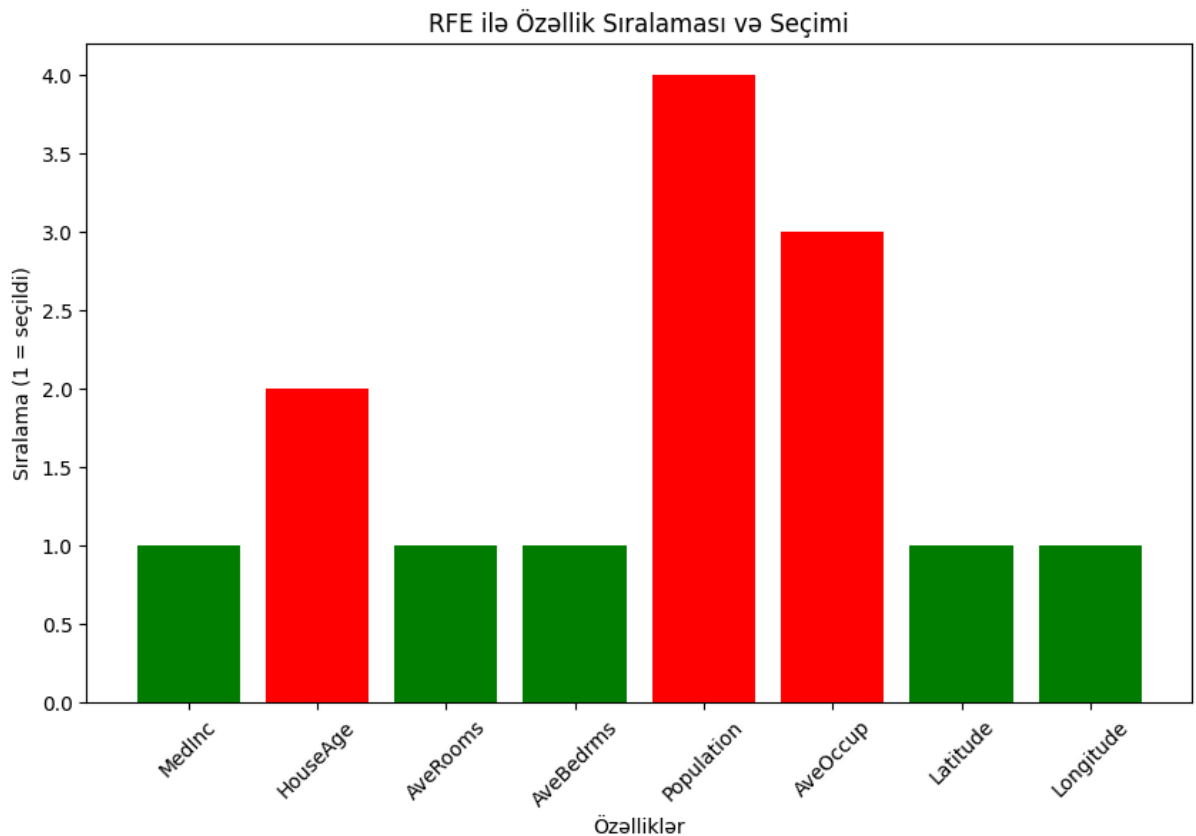
In []:

Seçilən Özəlliklər:

```
Index(['MedInc', 'AveRooms', 'AveBedrms', 'Latitude', 'Longitude'], dtype='object')
```

Özəllik Sıralaması (1=ən yaxşı):

```
MedInc      1
AveRooms    1
AveBedrms   1
Longitude   1
Latitude    1
HouseAge    2
AveOccup    3
Population  4
dtype: int64
```



5.3. Embedded Metodlar

Lasso (L1 Requlyarizasiyası)

- Reqressiya modellərində istifadə olunan bir requlyarizasiya metodudur.
- Xüsusiyyətlərin əmsallarını kiçildərək bəzilərini sifıra endirir.
- Əmsalı sifıra enən xüsusiyyətlər modelə daxil edilmir, beləcə təbii bir xüsusiyyət seçimi edilmiş olur.

Düstur (Lasso reqressiyanın məqsəd funksiyası):

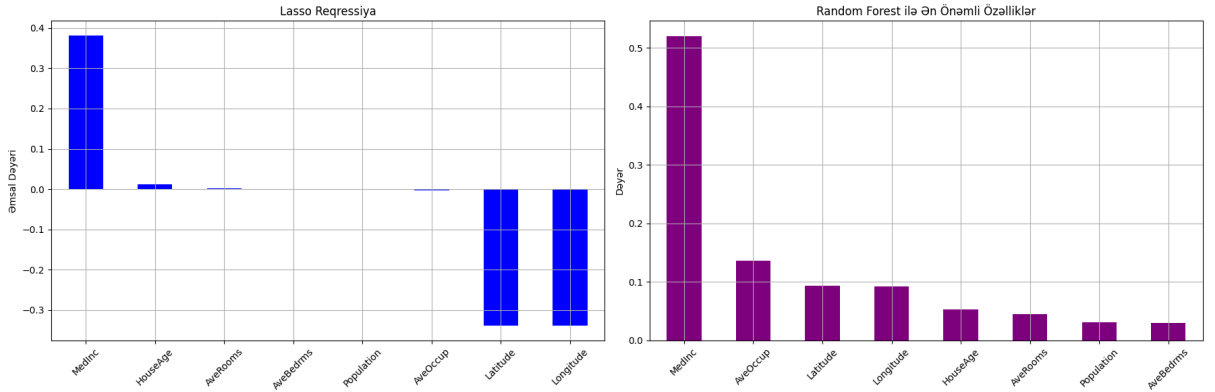
$$\min_{\beta} \left(\sum_{i=1}^n (y_i - X_i \beta)^2 + \lambda \sum_{j=1}^p |\beta_j| \right)$$

Burada λ requlyarizasiya parametridir.

Ağac Əsaslı (Random Forest, XGBoost)

- Ağac əsaslı modellər, xüsusiyyətlərin əhəmiyyətini avtomatik hesablayır.
- Xüsusiyyətlərin qərar ağacındakı budaqlanma tezliyi və informasiya qazanımı kimi metriklər əsas götürülür.
- Bu əhəmiyyət dərəcəsi, xüsusiyyət seçimində istifadə edilə bilər.

In []:



6: Deep Learningdə Ölçü Kiçiltməsi

6.1. Autoencoderlər

Autoencoder, giriş verilənlərini daha aşağı ölçülü gizli fəzaya (latent space) kodlayan və buradan orijinal veriləri yenidən yaratmağa çalışan süni neyron şəbəkəsidir. Bu quruluş sayəsində model verilərin əsas xüsusiyyətlərini sıxılmış şəkildə öyrənir.

Encoder – Decoder Quruluşu

- **Encoder:** Giriş verilənlərini alır və ölçüsünü azaldaraq latent vektor yaradır.
- **Decoder:** Bu latent vektordan orijinal veriləri yenidən yaratmağa çalışır.

Məqsəd, giriş və çıxış arasındakı fərqi minimuma endirərək veriləri mənalı şəkildə sıxmaqdır.

Latent Fəza (Gizli Fəza)

- Encoderin çıxışı olan aşağı ölçülü təmsil sahəsidir.
- Bu fəzadakı vektorlar, nümunələrin əsas xüsusiyyətlərini əks etdirir.
- Latent fəza, ölçü kiçiltməsi və feature selection üçün istifadə olunur.

Undercomplete vs Overcomplete Autoencoderlər

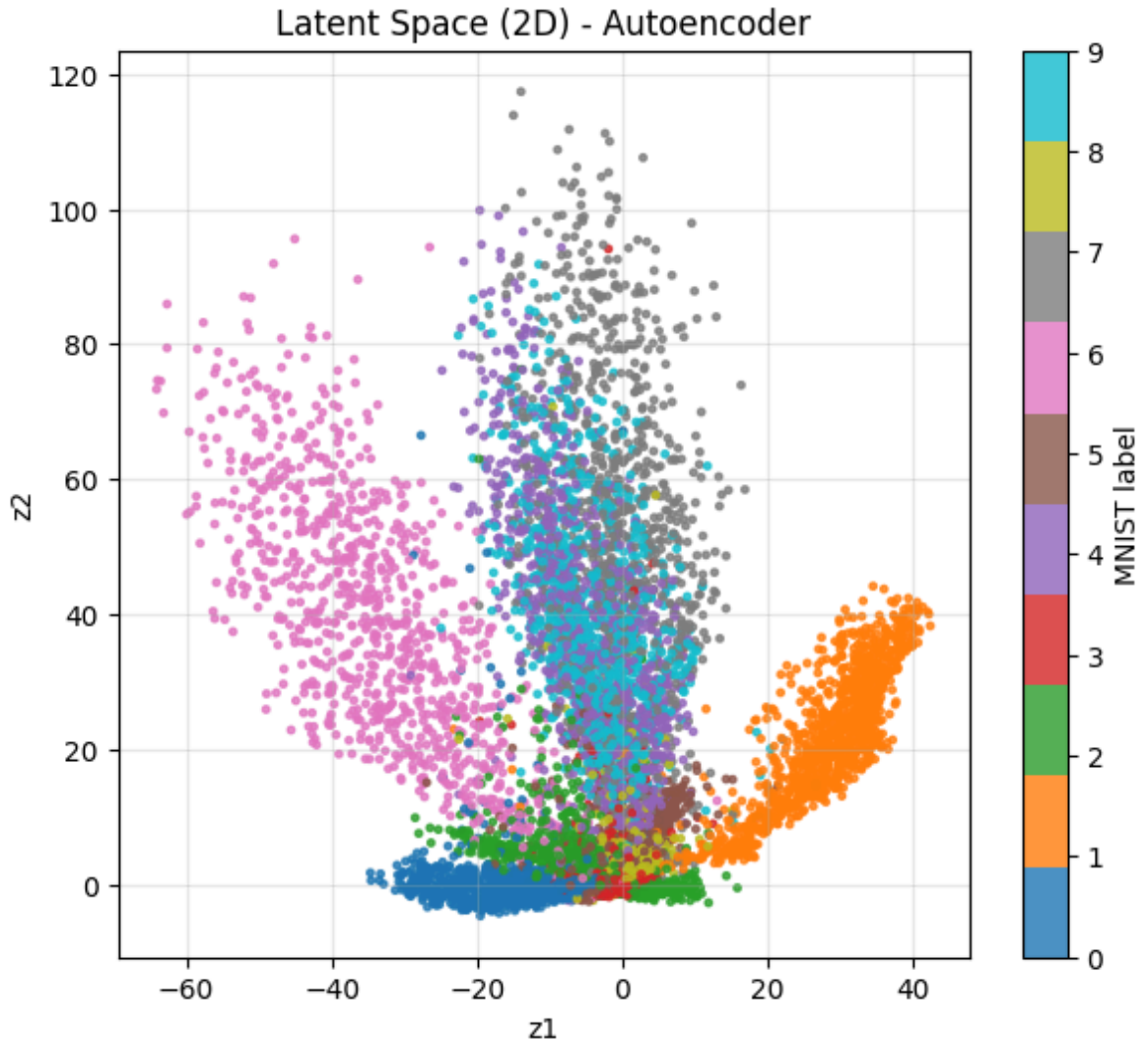
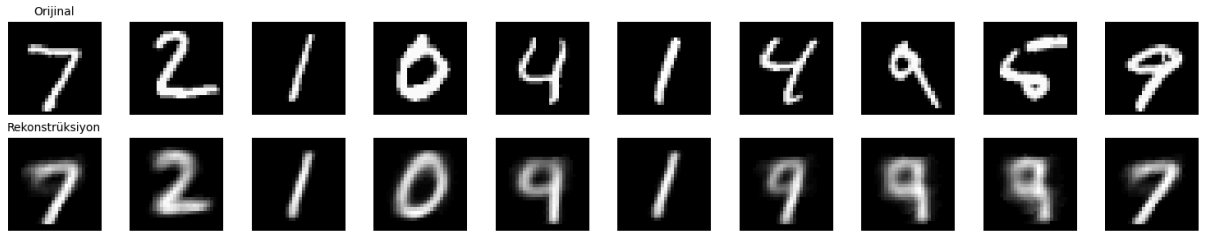
- **Undercomplete Autoencoder:** Gizli qatdakı neyronların sayı giriş ölçüsündən azdır. Bu halda model sıxılma edir və lazımsız məlumatları atır.
- **Overcomplete Autoencoder:** Gizli qat girişdən daha çox neyrona malikdir. Burada məqsəd sıxılma deyil, verilərdəki strukturları öyrənməkdir, lakin həddindən artıq öyrənmə (overfitting) riski artır.

Autoencoderlərin istifadə sahələri

- **Şəkil Sıxılması:** Yüksək keyfiyyətli tibbi görüntülər (MR, CT) kimi böyük datasetlərdə, autoencoderlər məlumatları sıxaraq saxlama və ötürmə xərclərini azaldır.
- **Anomaliyaların Aşkarı:** Sənaye maşınlarında sensor data autoencoder ilə modelləşdirilib normal vəziyyət öyrənilir. Yenidən yaratma xətası yüksəkdirsə, qəza və ya anomaliya kimi işarələnir.
- **Denoising:** Bulanıq və ya korlanmış görüntülər autoencoder ilə təmizlənərək orijinal görüntüyə oxşadılır.

In [3]:

313/313 ————— 0s 1ms/step
313/313 ————— 0s 882us/step



1/1 ————— 0s 42ms/step

Interp: 1 -> 7



6.2. Variational Autoencoders (VAE)

VAE, klassik autoencoderə probabilistik yanaşma gətirir. Hər data nöqtəsi üçün bir latent paylanması öyrənir və nümunələr bu paylanmadan çəkilərək yaradılır.

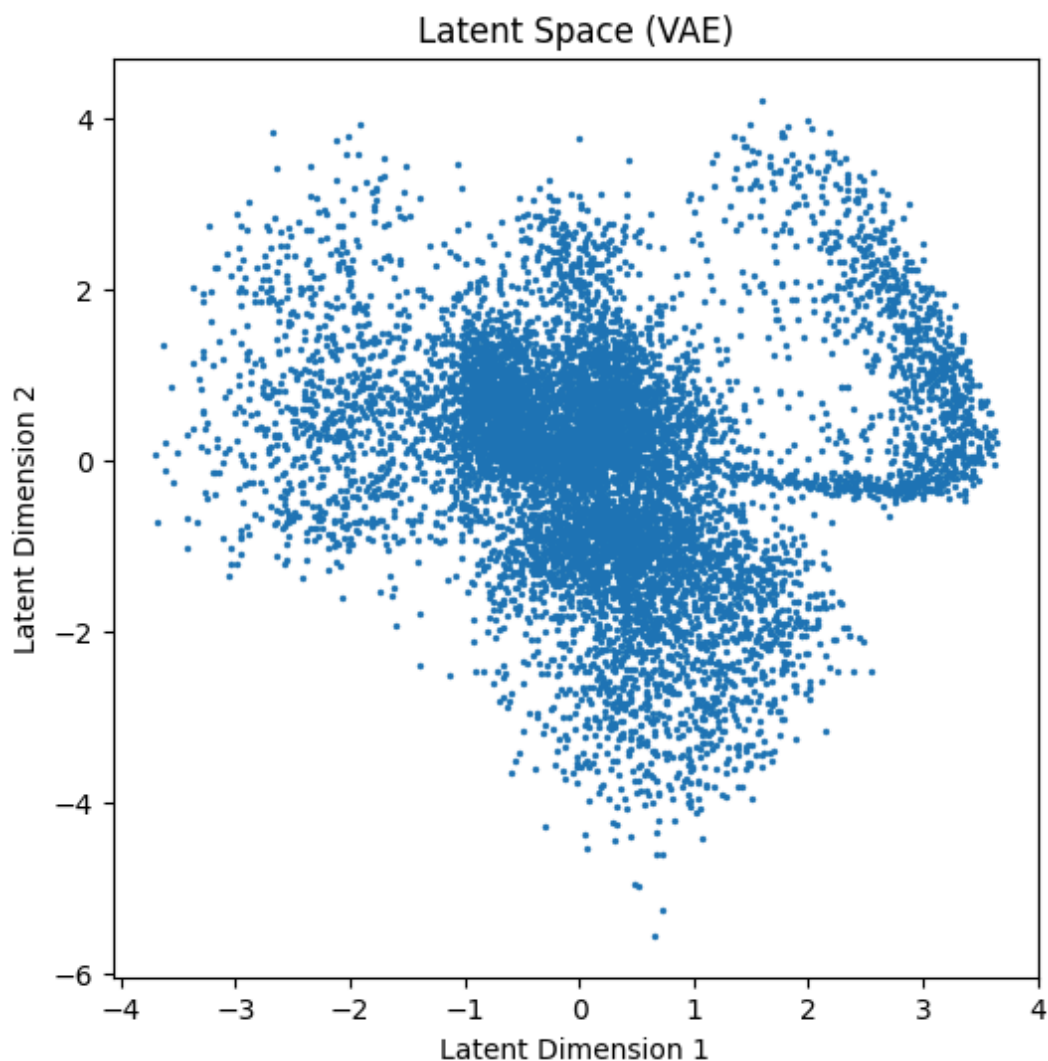
- Latent dəyişənlərin müəyyən bir paylanmaya (adətən normal paylanmaya) uyğun gəlməsi üçün model öyrədilir.
- Bu metod, daha hamar və davamlı latent fəza yaradır.
- VAE, verilərin yaradılması (generative modeling) və ölçü kiçiltməsi üçün güclü bir alətdir.

Variational Autoencoders (VAE) istifadə sahələri

- **Yeni Molekul Dizaynı:** Kimya və biologiyada, VAE-lər latent fəzada kimyəvi strukturları təmsil edərək yeni və orijinal molekullar yaradır.
- **Üz Yaradılması:** VAE-lər istifadə edilərək üz ifadəsi dəyişdirmə, üz yaratma kimi generativ modellər inkişaf etdirilir.
- **Data Augmentation:** Az verilər olan siniflər üçün VAE ilə oxşar, lakin yeni nümunələr yaradılaraq model train iningi dəstəklənir.

In [10]:

79/79 ————— 0s 2ms/step



6.3. Contrastive Learning və Representation Learning

Contrastive learning, oxşar nümunələri bir-birinə yaxın, fərqli nümunələri isə uzaq saxlamağı hədəfləyən bir texnikadır.

- Bununla model, dataların əsas quruluşunu ələ keçirən keyfiyyətli, fərqləndirici özəlliklər öyrənir.
- Bu özəlliklər ölçü kiçiltməsi üçün birbaşa istifadə edilə bilər.
- Contrastive learning, xüsusilə **unsupervised** öyrənmədə geniş yayılıb.

Contrastive Learning və Representation Learningin istifadə sahələri

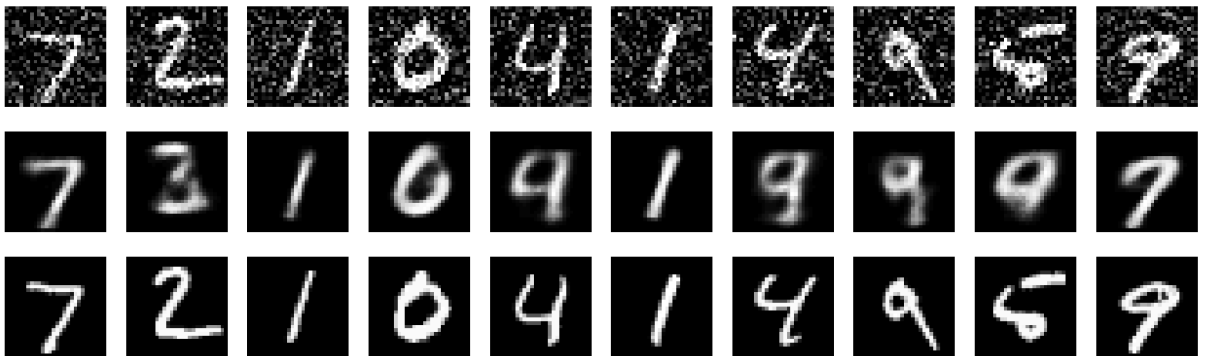
- **Şəkil Tanıma:** Böyük etiketlenməmiş vizual datasetlərdə, contrastive learning ilə keyfiyyətli feature extraction edilir; bu sayədə daha az etiketlə yüksək performans əldə edilir.
- **Səs Tanıma:** Danışiq datasetlərində oxşar səsləri bir-birinə yaxınlaşdırıb fərqli səsləri ayıraraq təmsil öyrənməsi edilir, beləliklə daha yaxşı təsnifat və tanıma təmin olunur.
- **Tövsiyə Sistemləri:** İstifadəçi davranışlarını latent təmsillərə çevirərək oxşar istifadəçiləri və məhsulları qruplaşdırmaqda istifadə olunur.

In [14]:

313/313

0s 1ms/step

Dense Autoencoder - Noisy Input, Denoised Output, Original



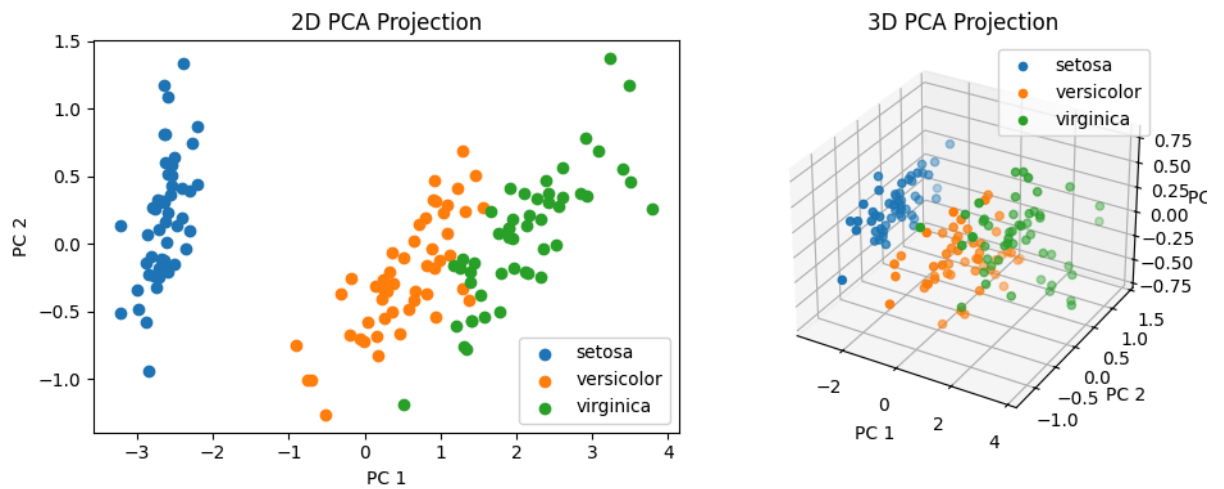
7: Tətbiq və Dəyərləndirmə

7.1 Ölçü Kiçiltməsi Sonrası Vizualizasiya (2D, 3D)

Ölçü kiçiltməsi metodlarının ən təməl tətbiqlərindən biri, yüksək ölçülü verilənləri **2D və ya 3D fəzada vizuallaşdıraraq** verilən strukturunu daha anlaşıqlı etməkdir. Xüsusilə PCA, t-SNE, UMAP kimi metodlar, datadakı qruplaşma, sinif fərqi və ya struktural xüsusiyyətləri vizual olaraq ortaya qoyur.

- **2D Scatter Plot:** Ən geniş yayılmış metoddur. İki əsas komponent və ya ölçü seçilir, hər data nöqtəsi bu müstəvidə göstərilir.
- **3D Scatter Plot:** Üç ölçülü proyeksiyalar daha çox məlumatı vizual olaraq daşıyır, lakin qrafiklə qarşılıqlı əlaqə tələb edir.

In [15]:



7.2 Model Performansına Təsiri

Ölçü kiçiltməsi əməliyyatı, maşın öyrənməsi modellərinin performansına həm müsbət, həm də mənfi təsir göstərə bilər.

- **Üstünlüklər:**

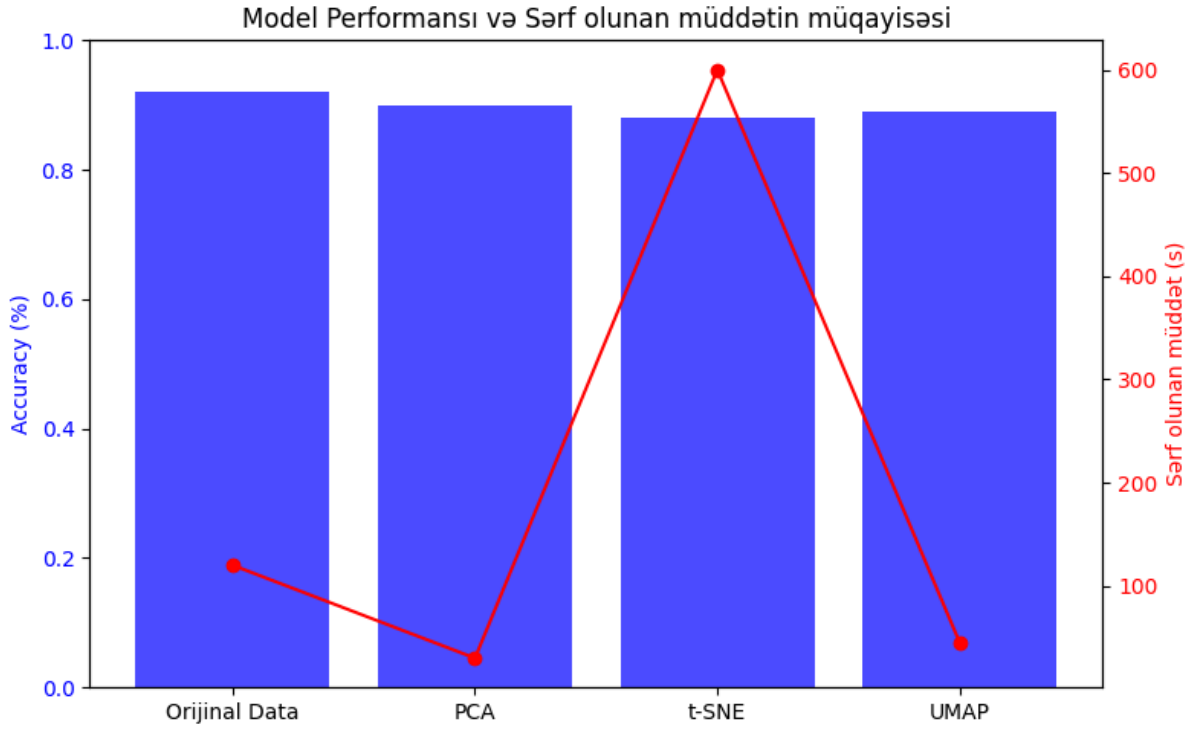
- Noise-i azaldaraq modeli sadələşdirir.
- Hesablama vaxtını və yaddaş istifadəsini azaldır.
- Həddindən artıq öyrənmənin (overfitting) qarşısını almağa kömək edə bilər.

- **Çatışmazlıqlar:**

- Məlumat itkisinə səbəb ola bilər, xüsusilə yüksək ölçülü verilənlərdə əhəmiyyətli xüsusiyyətlər nəzərdən qaça bilər.
- Bəzi hallarda performans azala bilər.

Müqayisə üçün model doğruluq dəyərləri ölçü kiçiltməsi əvvəl və sonra araşdırılmalıdır.

In [20]:



7.3 Məlumat itkisi və Məlumatın Qorunması

Ölçü kiçiltməsi əməliyyatı, orijinal məlumatların hamısını saxlamaya bilər. Əsas sual: **Nə qədər məlumat itkisi yaşanır?**

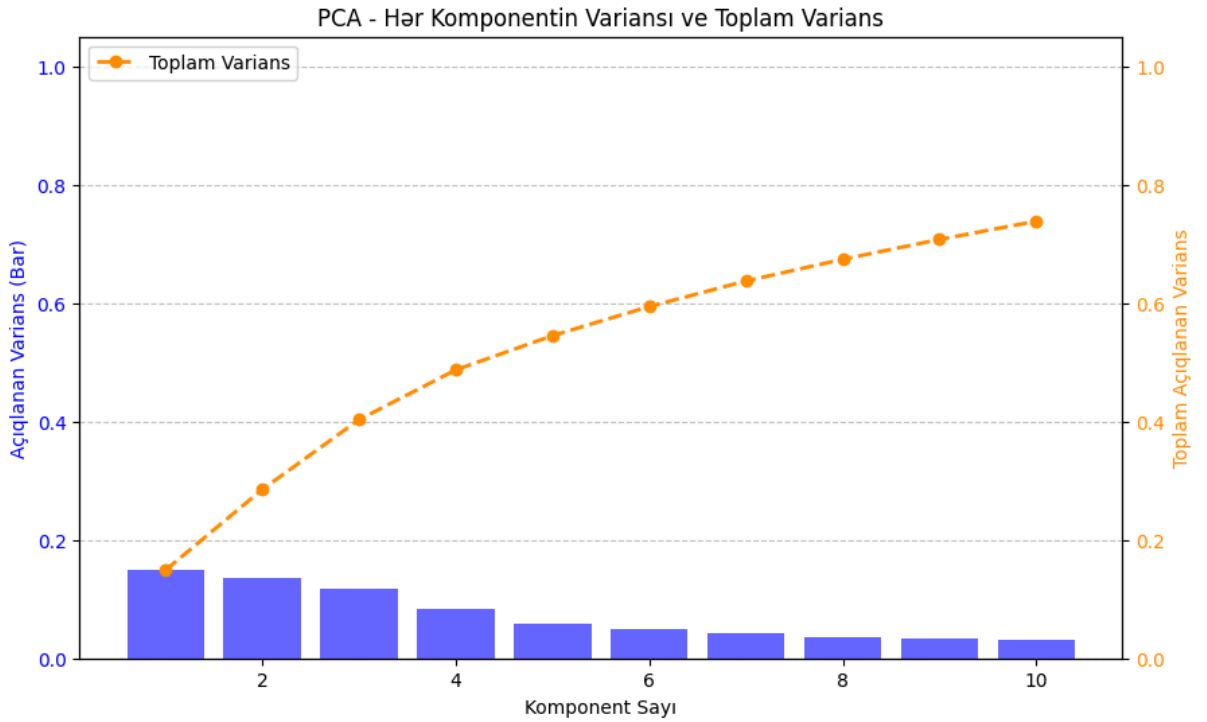
- PCA'da qorunan dispersiya nisbəti aşağıdakı kimi hesablanır:

$$\text{Qorunan Dispersiya} = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^d \lambda_i}$$

Burada λ_i i'nci eigenvaluedir, k seçilən ölçü sayı, d orijinal ölçü.

- t-SNE, UMAP kimi metodlarda isə adətən vizual keyfiyyət və qonşuluq əlaqəsi müşahidə edilir.
- Yenidən yaratma xətası (xüsusilə autoencoder modellərdə) məlumat itkisini kəmiyyət olaraq ölçür.

In [29]:



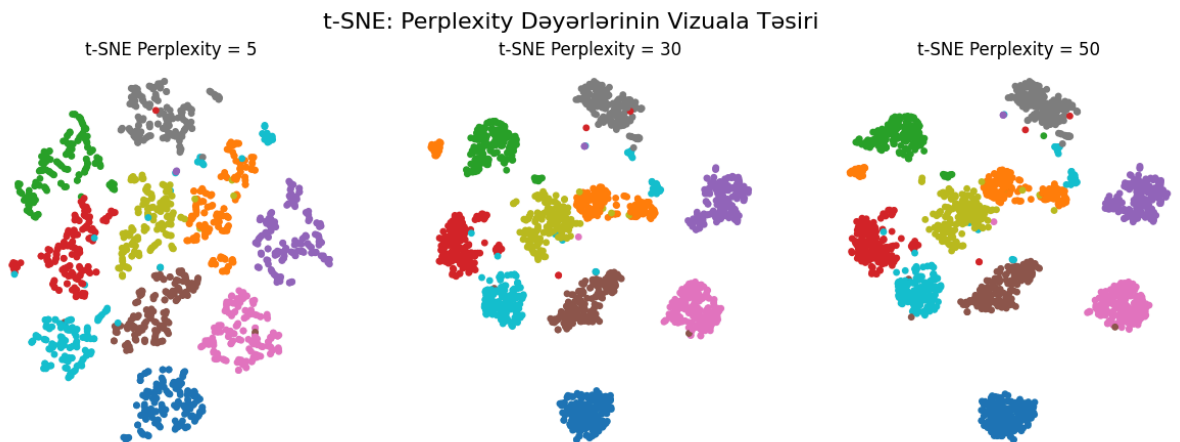
7.4 Hiperparametrlər və Overfitting Riski

Ölçü kiçiltməsi alqoritmlərində hiperparametr seçimi nəticələrə birbaşa təsir göstərir:

- **PCA:** Komponent sayı k seçimi kritikdir. Çox az komponent məlumat itkisinə, çox komponent isə lazımsız mürəkkəbliyə səbəb olur.
- **t-SNE:**
 - *Perplexity* parametri, məlumat sıxlığını təsir edir. Çox kiçik və ya çox böyük dəyərlər pis proyeksiyalar yarada bilər.
 - *Learning rate* optimizasiyanın sürətinə təsir edir.
- **UMAP:** Qonşu sayı və məsafə metriki parametrləri vacibdir.
- **Autoencoder:** Gizli qat ölçüləri, öyrənmə sürəti, dropout kimi ayarlar həddindən artıq öyrənmənin qarşısını almaq üçün optimallaşdırılmalıdır.

Overfitting riski, xüsusilə supervised ölçü kiçiltməsi (LDA kimi) və autoencoderlərdə ortaya çıxır. Düzgün validation və early dropout metodları istifadə edilməlidir.

In [31]:



7.5 Vaxt və Yaddaş Mürəkkəbliyi

Ölçü kiçiltməsi metodlarının hesablama xərcləri fərqlidir:

| Metod | Vaxt Mürəkkəbliyi | Yaddaş Mürəkkəbliyi | Qeydlər |
|-------------|------------------------------|------------------------|--|
| PCA | $O(nd^2 + d^3)$ (SVD əsaslı) | $O(d^2)$ | Böyük veri setlərində optimallaşdırıla bilər |
| t-SNE | Təxminən $O(n^2)$ | Yüksək | Yavaş, böyük veri üçün çətindir |
| UMAP | $O(n \log n)$ | Orta | Daha sürətli və miqyaslanabilir |
| Autoencoder | GPU ilə sürətləndirilir | Model ölçüsündən asılı | Təlim müddəti parametrlə bağlıdır |

Böyük veri setlərində metod seçimi və hardware infrastrukturunu nəzərə alınmalıdır.

8: İrəli Səviyyə Mövzular

8.1. Supervised və Unsupervised Ölçü Kiçiltməsi

Ölçü kiçiltməsi texnikaları, verilərə və problemə görə **supervised** və ya **unsupervised** ola bilər:

- **Unsupervised Metodlar:** Məlumat etiketlərinə baxmadan, yalnız xüsusiyyətlər arasındakı quruluşu istifadə edərək ölçü kiçiltməsi edirlər.

Məsələn: PCA, t-SNE, UMAP. Məqsəd adətən verilərin ümumi strukturunu, dispersiyasını və ya qruplaşmasını qorumaqdır.

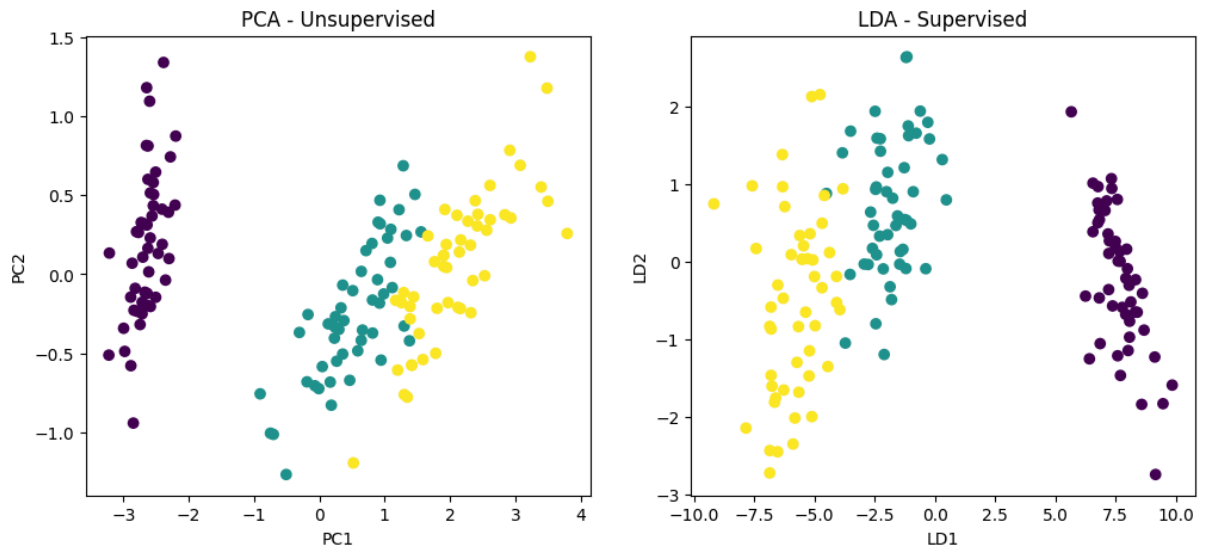
- **Supervised Metodlar:** Məlumat ilə birlikdə gələn etiket məlumatlarını istifadə edərək, siniflər və ya hədəf dəyişən istiqamətində ölçü kiçiltməsi edirlər.

Məsələn: Linear Discriminant Analysis (LDA), Supervised PCA, Neighborhood Components Analysis (NCA)

Məqsəd, sinif fərqi və ya hədəf proqnozunun dəqiqliyini artıracaq ölçü kiçiltməsi etməkdir.

Nümunə: Bir təsnifat problemi üçün LDA, sinif fərqi maksimumlaşdıraraq oxları yaradır. Ancaq PCA sadəcə data paylanması üçün ən çox dispersiya göstərdiyi oxları tapır.

In [33]:



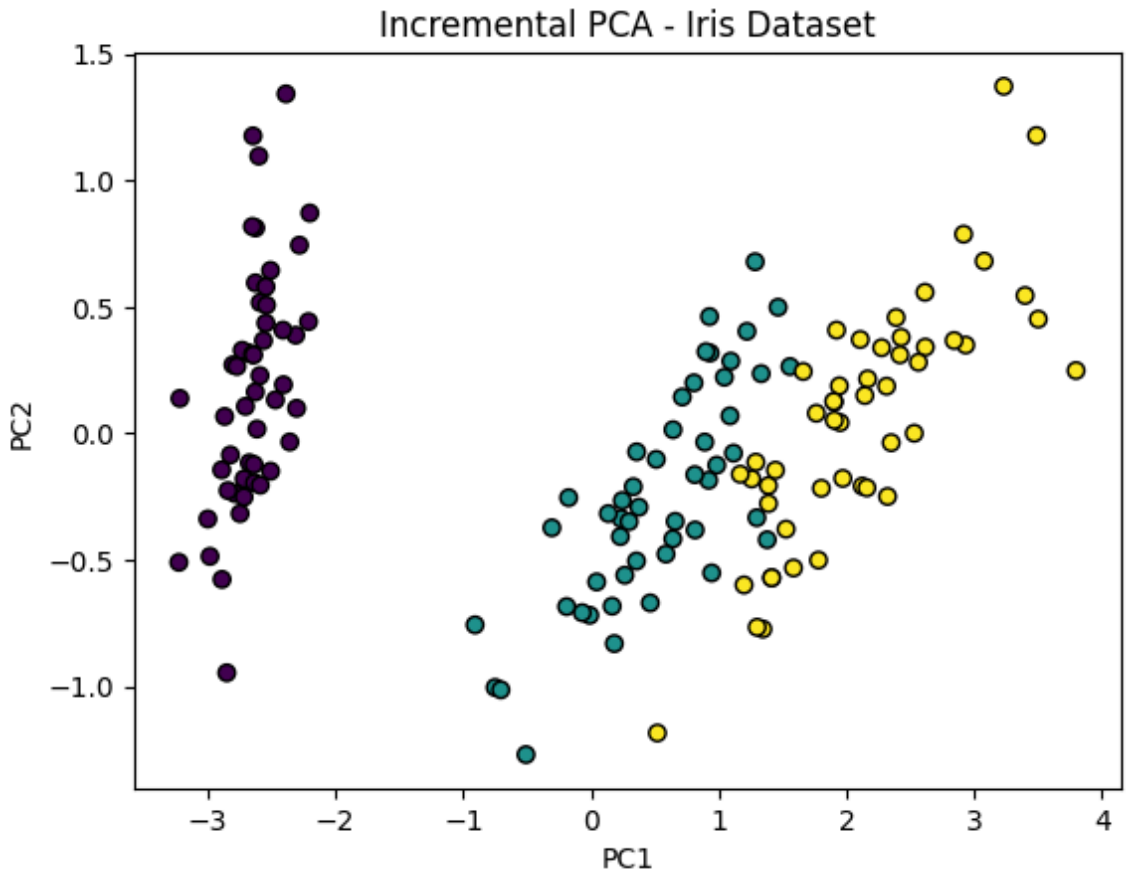
8.2. Online / Incremental PCA

PCA ümumiyyətlə bütün dataset yaddaşa yükləndikdən sonra hesablanır, bu da böyük datasetlərdə praktik olmayabilir.

- **Incremental PCA (IPCA)**, məlumatları kiçik hissələr (batch) şəklində emal edir, yaddaşa bütün data yerləşmədiyi halda və ya axın streaming datalarda istifadə olunur.
- Xüsusilə real vaxt sistemlərində, davamlı yeni data gəldikcə modeli yeniləmək üçün uyğundur.

Riyazi olaraq: Data bloklar şəklində emal olunur, aralıq nəticələr yenilənir və yekun komponentlər hesablanır.

In [42]:



8.3. Sparse PCA

Standart PCA'da hər komponent adətən bütün xüsusiyyətlərin xətti kombinasiyasıdır. Bu, komponentlərin şərhini çətinləşdirə bilər.

- **Sparse PCA**, komponentlərdəki əmsalları seyrəkləşdirir (çox sayda sifıra yaxın dəyər olur).
- Beləcə komponentlər daha az sayda xüsusiyyət ehtiva edir, şərh edilmə qabiliyyəti artır.

İstifadə Sahəsi: Xüsusilə çox yüksək ölçülü və noised datasetlərdə, məsələn, gen expression, text mining.

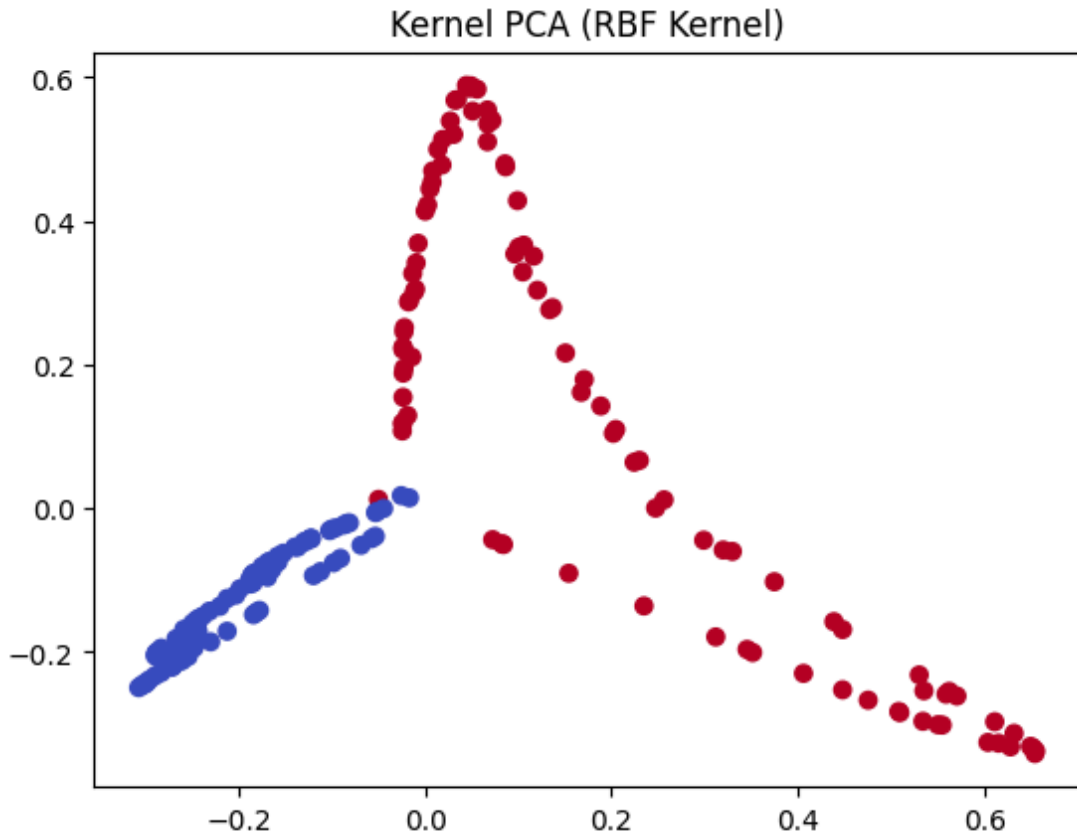
8.4. Kernel PCA

PCA sadəcə xətti əlaqələri tuta bilər. Real datalar çox vaxt qeyri-xətti quruluşlar ehtiva edir.

- **Kernel PCA**, dataları yüksək ölçülü bir fəzaya **kernel funksiyaları** ilə çevirir.
- Orada xətti PCA tətbiq edilir və beləcə orijinal fəzada qeyri-xətti ölçü kiçiltməsi edilmiş olur.

Populyar kernel funksiyaları: RBF (Gaussian), Polynomial, Sigmoid.

In [36]:



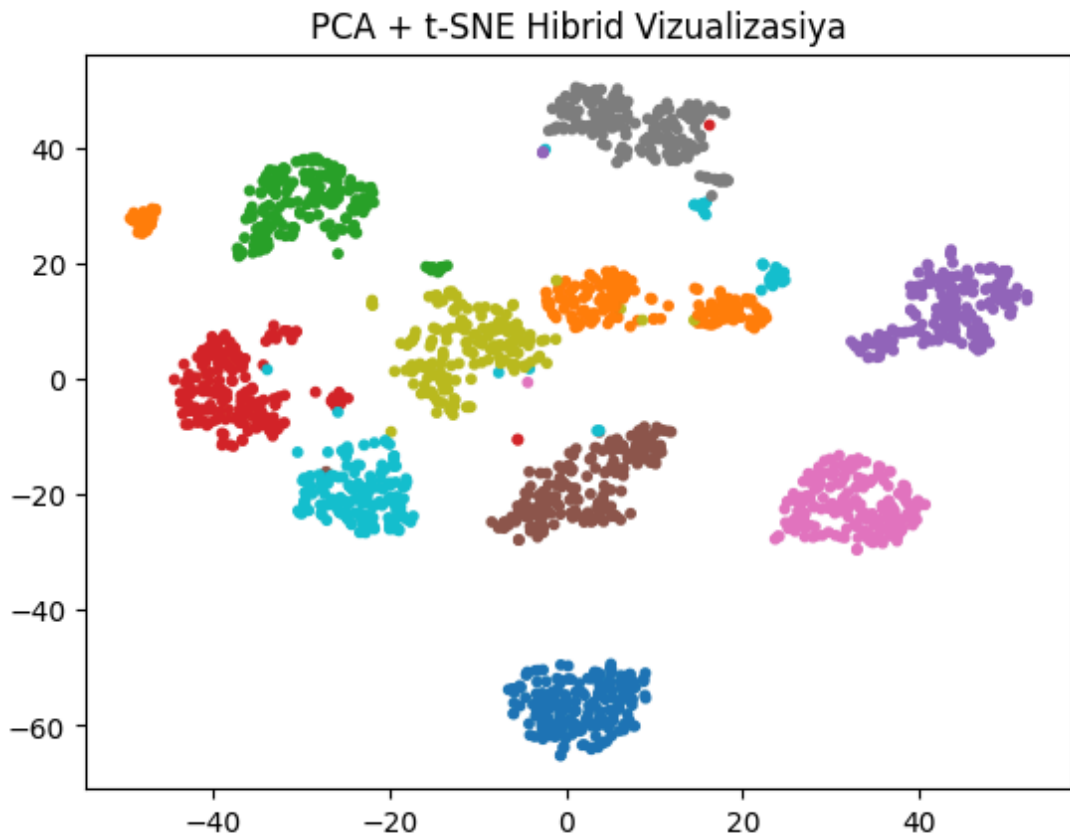
8.5. Metodların Birləşməsi (Hibrid Modellər)

Ölçü kiçiltməsi metodları bir-birini tamamlayıcı olaraq istifadə edilə bilər.

- Məsələn, əvvəlcə PCA ilə ölçü kiçiltməsi edib sonra t-SNE ilə vizualizasiya,
- Və ya Sparse PCA + LDA ilə həm şərh edilə bilən, həm də sinif ayrımı güclü komponentlər əldə etmək.

Bu yanaşmalar, mürəkkəb datasetlərdə elastiklik və performansını təmin edir.

In [37]:



8.6. Zaman Seriyası və Mətn üçün Ölçü Kiçiltməsi

Zaman seriyası və mətn datasetləri xüsusi quruluşlar ehtiva edir, bu səbəbdən klassik metodlar yetərli olmaya bilər.

- **Zaman Seriyası:**

Sliding window və feature extraction + PCA/IPCA, Avto reqressiv modellərlə birlikdə ölçü kiçiltməsi, RNN və Autoencoder əsaslı dərin öyrənmə metodları.

- **Mətn:**

TF-IDF, Word2Vec, BERT kimi embedding texnikaları sonrası PCA, t-SNE, UMAP, Mövzu Modelləşdirməsi (LDA - Latent Dirichlet Allocation) ilə mövzu ölçü kiçiltməsi.

In [39]:

Sliding Window + PCA ilə Time Series Ölçü Kığıltması

