PYTHON PROGRAMMING AND MACHINE LEARNING

FEATURE ENGINEERING AND REDUCTION

Yunghans Irawan (yirawan@nus.edu.sg)

Objectives

- Understand the importance of feature engineering in machine learning

- Able to perform some common feature engineering and dimensionality reduction technique

- Understand the basic concepts of:
  - PCA
  - LDA

Feature Engineering

- The usefulness and accuracy of our machine learning model is greatly influence by the features in our data

- Data collection and pre-processing takes up a significant portion of effort in a machine learning project

Recap of Common Data Preparations

- Categorical Variable
  - Encode into numeric value (integer)
- One-Hot-Encoding
  - Convert each category into its own column with 0 and 1 value
  - 1 means that sample belong to that category
- Binning
  - Sometimes converting a continuous number into category gives us better model
    - E.g. salary into salary range (<50k, 50-100k, 100-150k)

Recap of Common Data Preparations

- Missing Data
  - Exclude the features if there's too many missing data
  - Replace blank with a default value (average value or a reasonable default)
    - Depends on the domain
- Unreasonable Data
  - Example: negative age
  - Exclude the rows or replace with default values

Feature Selection & Reduction

- Why
  - Visualization
  - Curse of Dimensionality
- How
  - Feature Selection
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)

Feature dimensions

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	car_name
0	18.0	8	307.0	130.0	3504.0	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693.0	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436.0	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433.0	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449.0	10.5	70	1	ford torino
5	15.0	8	429.0	198.0	4341.0	10.0	70	1	ford galaxie 500
6	14.0	8	454.0	220.0	4354.0	9.0	70	1	chevrolet impala
7	14.0	8	440.0	215.0	4312.0	8.5	70	1	plymouth fury iii
8	14.0	8	455.0	225.0	4425.0	10.0	70	1	pontiac catalina
9	15.0	8	390.0	190.0	3850.0	8.5	70	1	amc ambassador dpl
10	15.0	8	383.0	170.0	3563.0	10.0	70	1	dodge challenger se
11	14.0	8	340.0	160.0	3609.0	8.0	70	1	plymouth 'cuda 340
12	15.0	8	400.0	150.0	3761.0	9.5	70	1	chevrolet monte carlo
13	14.0	8	455.0	225.0	3086.0	10.0	70	1	buick estate wagon (sw)

What is the feature dimension?

High dimensional features

37 x 50 pixels = 1850 features!

```python
import pandas as pd

df = pd.DataFrame(lfw.data)
df.head()
```

	0	1	2	3	4	5	6	7	8	9	...
0	254.000000	254.000000	251.666672	240.333328	185.333328	144.000000	174.000000	196.666672	196.000000	192.333328	...
1	39.666668	50.333332	47.000000	54.666668	99.000000	120.666664	139.666672	157.666672	171.000000	177.666672	...
2	89.333336	104.000000	126.000000	141.333328	152.000000	155.333328	155.333328	160.000000	163.000000	166.666672	...
3	16.666666	7.666667	7.000000	6.000000	16.333334	70.000000	170.000000	169.666672	161.000000	106.333336	...
4	122.666664	121.000000	126.666664	129.333328	129.333328	134.666672	142.000000	142.666672	147.333328	152.000000	...

5 rows × 1850 columns

Visualization

Plotting features helps us find patterns

But, datasets typically have >=3 features

… just look at UCI repository

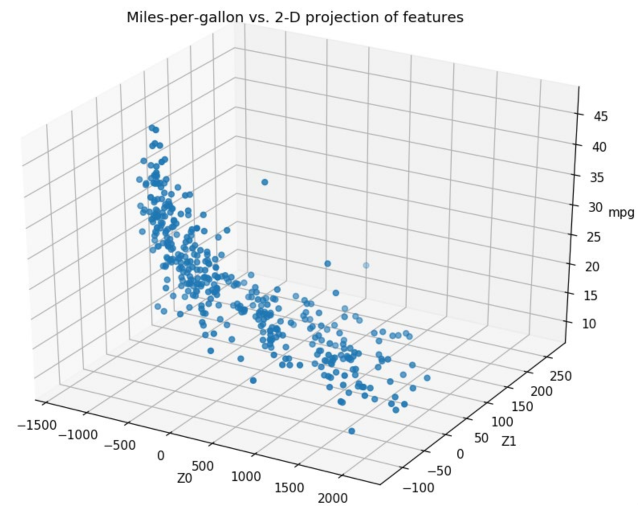
Problem: Humans can't see more than 3-D

Visualization

- Visualize to see relationships of features (X) with mpg (y)

y X

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	car_name
0	18.0	8	307.0	130.0	3504.0	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693.0	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436.0	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433.0	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449.0	10.5	70	1	ford torino
5	15.0	8	429.0	198.0	4341.0	10.0	70	1	ford galaxie 500
6	14.0	8	454.0	220.0	4354.0	9.0	70	1	chevrolet impala
7	14.0	8	440.0	215.0	4312.0	8.5	70	1	plymouth fury iii
8	14.0	8	455.0	225.0	4425.0	10.0	70	1	pontiac catalina
9	15.0	8	390.0	190.0	3850.0	8.5	70	1	amc ambassador dpl
10	15.0	8	383.0	170.0	3563.0	10.0	70	1	dodge challenger se
11	14.0	8	340.0	160.0	3609.0	8.0	70	1	plymouth 'cuda 340
12	15.0	8	400.0	150.0	3761.0	9.5	70	1	chevrolet monte carlo
13	14.0	8	455.0	225.0	3086.0	10.0	70	1	buick estate wagon (sw)



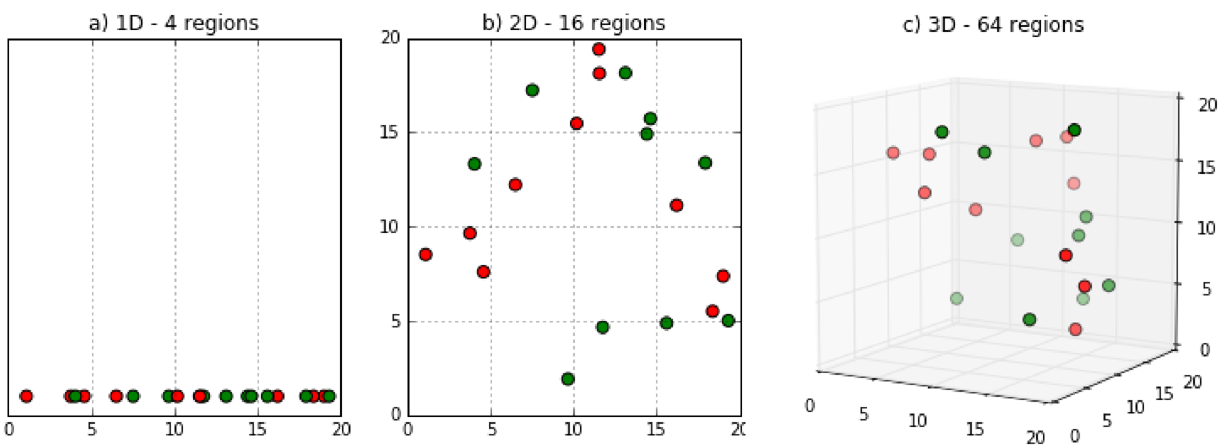
Original **X** (7-dimension) vs y

Plot **Z** (2-dimension) vs **y**

Curse of Dimensionality

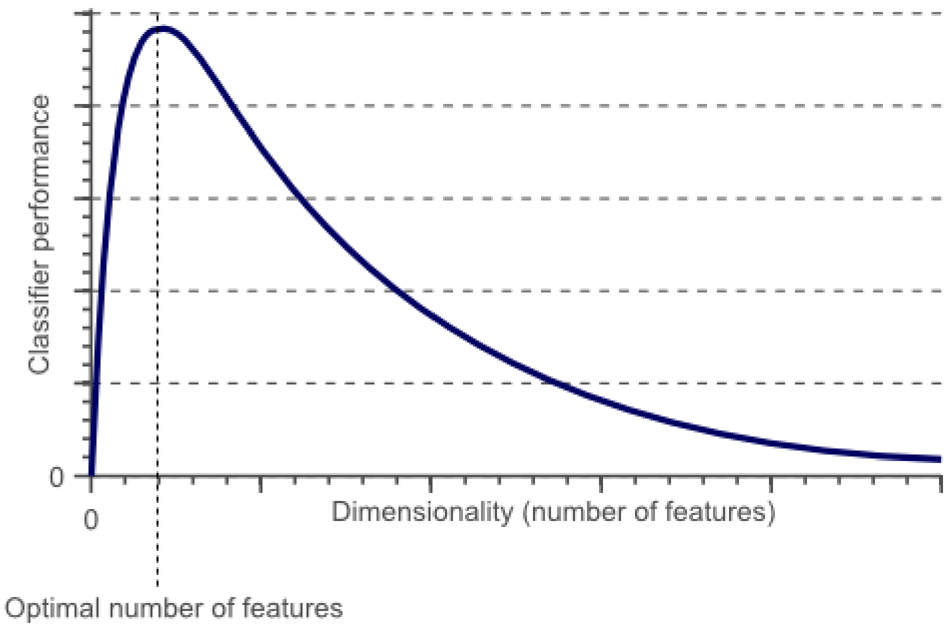
Machine learning is a **search (i.e. optimization) problem**

The search space **increases exponentially** with more features



Features	Regions
1	4
2	4^2
k	4^k

Dimensionality vs. model performance



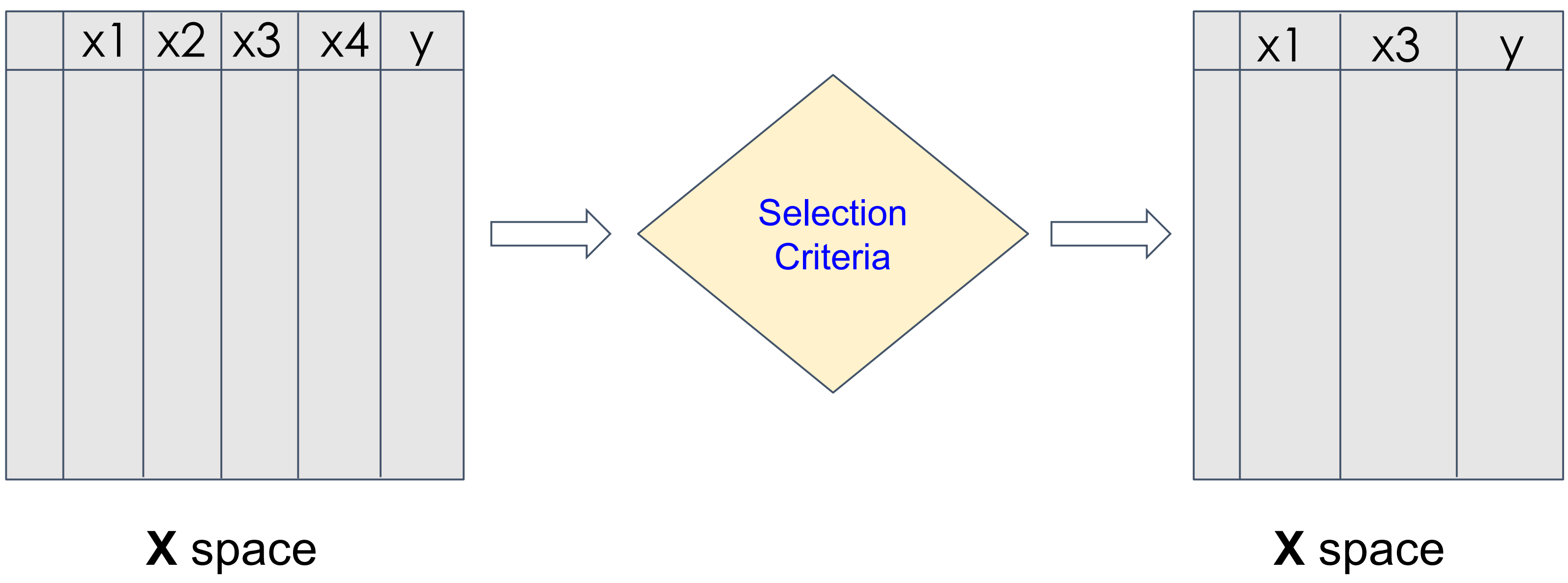
Techniques

Feature Selection

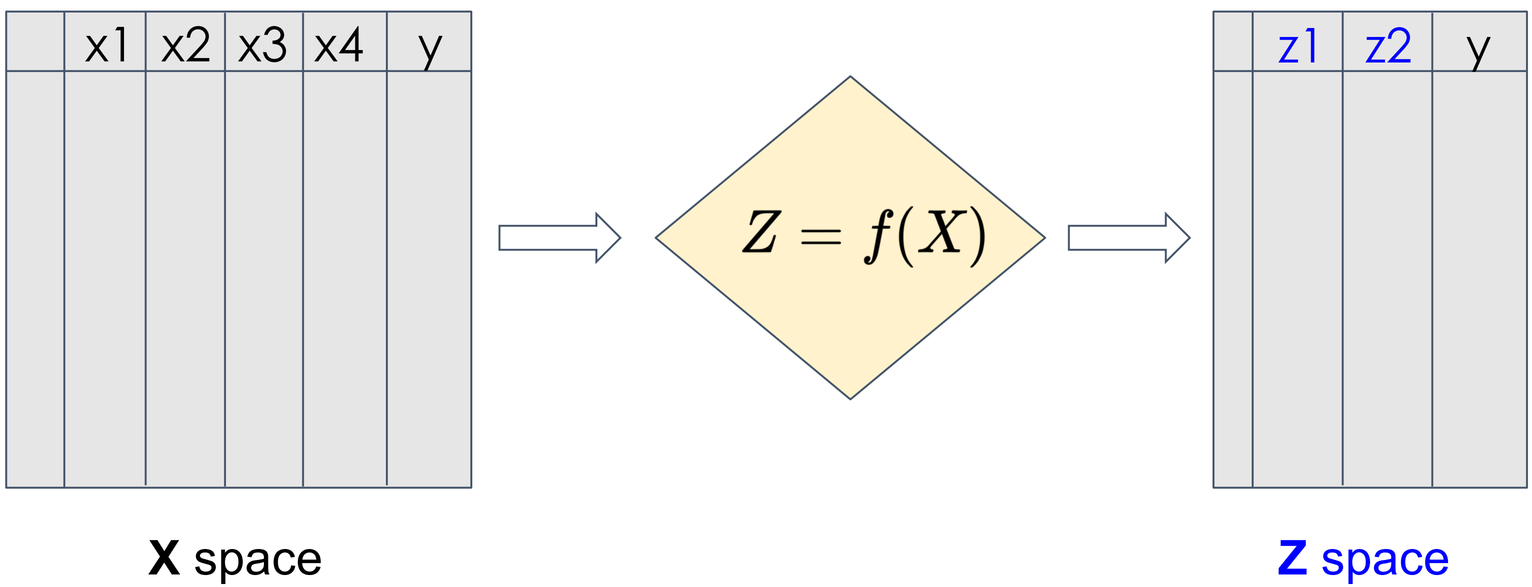
Principal Component Analysis (PCA)

Linear Discriminant Analysis (LDA)

Feature Selection



Feature Reduction



Feature Selection

- Ignore features that don't contribute much to the model
- Correlation
  - Too low with y => not much use
  - Too high with other features => redundant
- Statistical Tests
  - E.g. feature doesn't change very much (low variance)
  - [sklearn.feature_selection.VarianceThreshold]()
  - SelectKBest

Correlation

- Two questions:
  1. How related is each x with y?
     - Hint on what kind of features to use

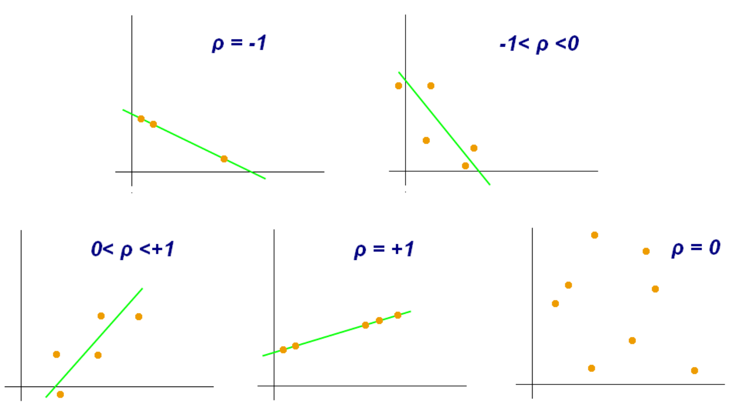
  2. How related is x1 with x2?
     - Including both adds more noise to model

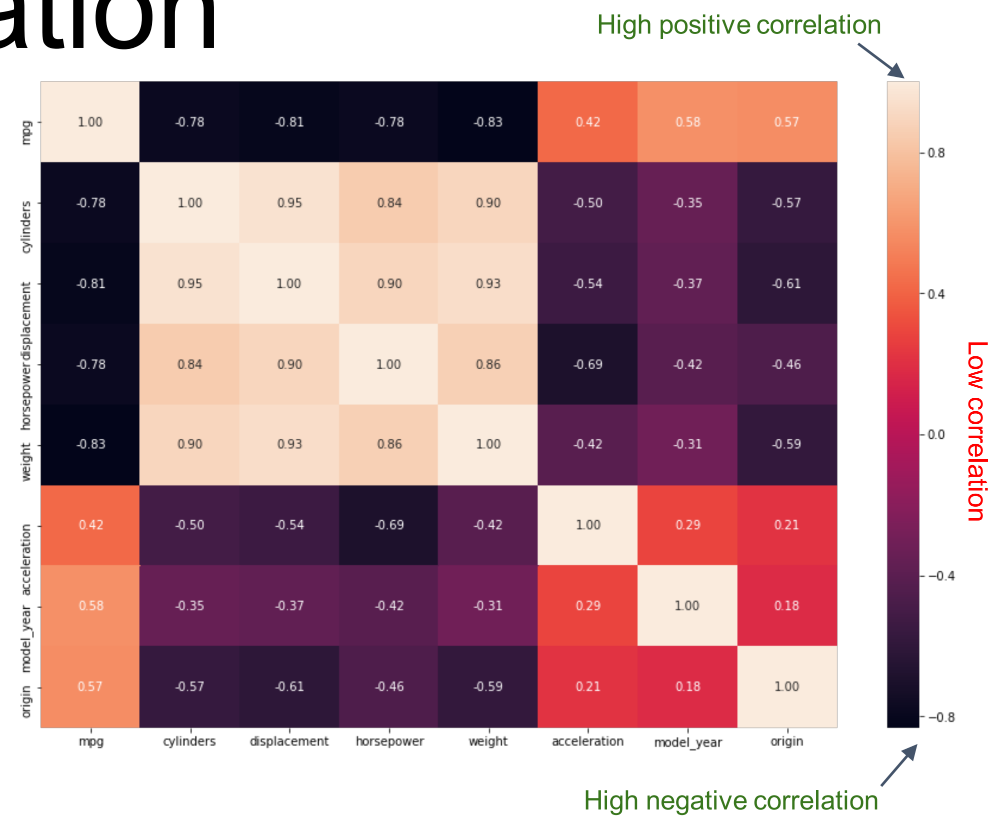
Pearson Correlation

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

where:

- n is the sample size
- x_i, y_i are the individual sample points indexed with *i*
- $\bar{x} = \frac{1}{n}\sum_{i-1}^{n} x_i$ (the sample mean); and analogously for $\bar{y}$



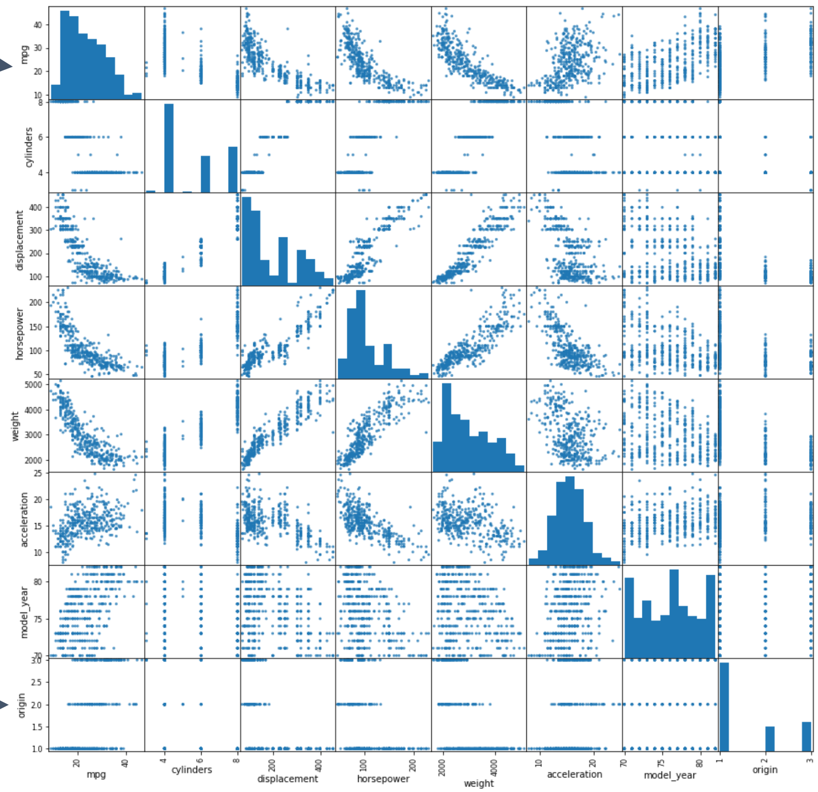


Scatter Matrix

Histogram

Scatter plot

Categorical data

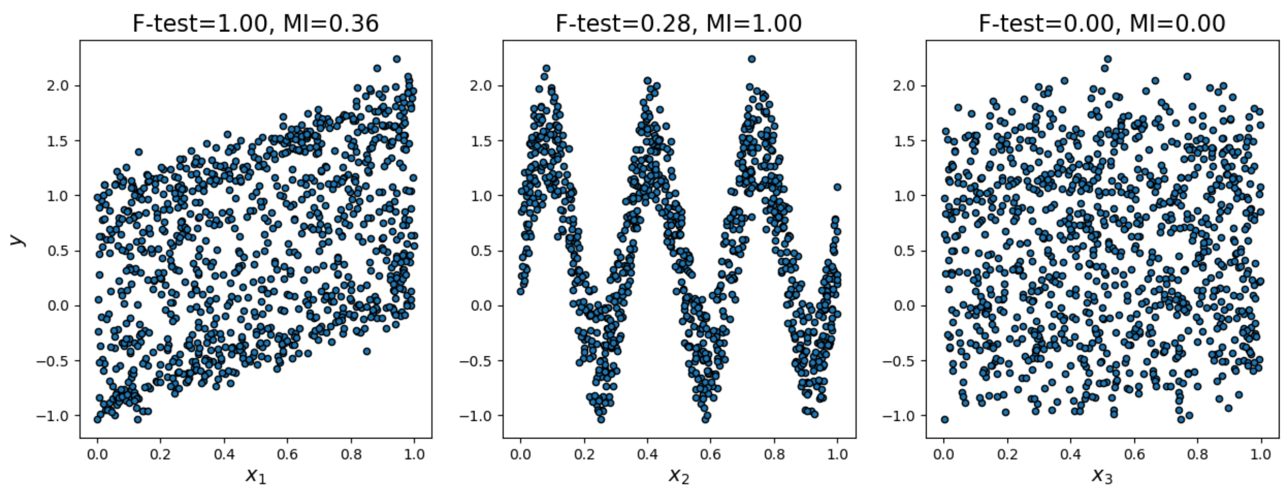


Large number of features

- Scatter Matrix or Correlation plots become hard to view properly
- Experiment programmatically:
  - Compute Pearson Correlation
  - Apply thresholding
    - For Regression tasks, can drop features where target correlation is below threshold
    - Can drop feature where intra-feature correlation is above threshold

Select K Best

- Select features based on F-test or Mutual Information
- Higher F-value means higher dependency between each X column and y



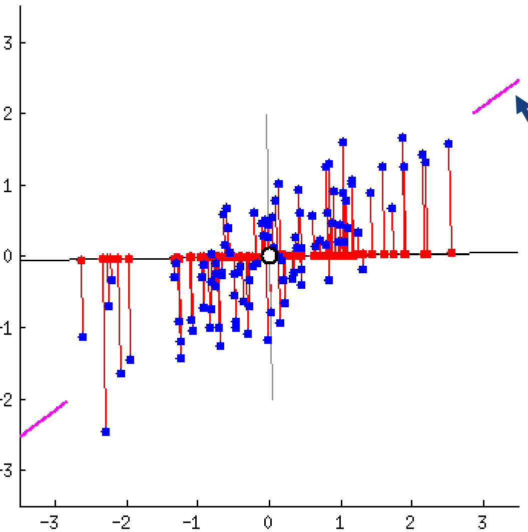
Principal Component Analysis (PCA)

- Find an orthogonal projection into a lower dimensional space

- Given X (n-dim) to Z (k-dim), where n > k:
- Finds Z-axes that capture the **highest variance for X**
- k subset of the principal components

PCA: Intuition

- Project from 2-D space to 1-D space



Pink line captures the highest variance of the data points

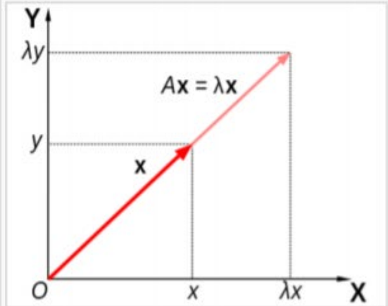
Eigenvector

- An eigenvector of a square matrix $\boldsymbol{A}$ is a non-zero vector $\boldsymbol{v}$ such that multiplication by $\boldsymbol{A}$ only changes the scale of $\boldsymbol{v}$

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$$

  – The scalar λ is known as eigenvalue

- If $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{A}$, so is any rescaled vector $s\boldsymbol{v}$. Moreover $s\boldsymbol{v}$ still has the same eigen value. Thus look for a unit eigenvector



Matrix A acts by stretching the vector x, not changing its direction, so x is an eigenvector of A.

Wikipedia

Non-examinable

Example of Eigenvalue/Eigenvector

- Consider the matrix $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$
- Taking determinant of $(A\text{-}\lambda I)$, the char poly is

$$\boxed{|A - \lambda I| = \begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix} = 3 - 4\lambda + \lambda^2}$$

- It has roots $\lambda=1$ and $\lambda=3$ which are the two eigenvalues of A
- The eigenvectors are found by solving for $\boldsymbol{v}$ in $A\boldsymbol{v}=\lambda\boldsymbol{v}$, which are $\boxed{v_{\lambda=1} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, v_{\lambda=3} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}}$

Non-examinable

SVD Definition

- Write A as a product of 3 matrices: $A = UDV^{\mathrm{T}}$
  - If A is $m \times n$, then U is $m \times m$, D is $m \times n$, V is $n \times n$
- Each of these matrices have a special structure
  - U and V are orthogonal matrices
  - D is a diagonal matrix not necessarily square
    - Elements of Diagonal of D are called *singular values of* A
    - Columns of U are called *left singular vectors*
    - Columns of V are called *right singular vectors*
- SVD interpreted in terms of *eigendecomposition*
  - Left singular vectors of A are eigenvectors of AA^{T}
  - Right singular vectors of A are eigenvectors of $A^{\mathrm{T}}A$
  - Nonzero singular values of A are square roots of eigen values of $A^{\mathrm{T}}A$. Same is true of AA^{T}

Non-examinable

PCA (using SVD)

Singular Value Decomposition

1. Subtract mean
2. Compute C, the Covariance Matrix
3. Use C to compute eigenvectors and eigenvalues
4. Sort eigenvectors by decreasing eigenvalues and choose k largest. These are the principal components
5. Transform X using the principal components

Reference: A Tutorial on Principal Component Analysis

Non-examinable

Mathematics

Covariance Matrix

$$C = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ v_{n1} & \dots & \dots & v_{nn} \end{bmatrix}$$

$$v_{ab} = \sum \frac{(x_a - \mu_a)(x_b - \mu_b)}{n}$$

$$\det(A - \lambda I) = 0$$

$$\det\left(\begin{pmatrix} 504 & 360 & 180 \\ 360 & 360 & 0 \\ 180 & 0 & 720 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\right)$$

$$-\lambda^3 + 1584\lambda^2 - 641520\lambda + 25660800$$

$$\lambda \approx 44.81966\ldots, \lambda \approx 629.11039\ldots, \lambda \approx 910.06995\ldots$$

$$\mathbf{A}v = \boldsymbol{\lambda} v \qquad \begin{pmatrix} -3.75100\ldots \\ 4.28441\ldots \\ 1 \end{pmatrix}, \begin{pmatrix} -0.50494\ldots \\ -0.67548\ldots \\ 1 \end{pmatrix}, \begin{pmatrix} 1.05594\ldots \\ 0.69108\ldots \\ 1 \end{pmatrix}$$

https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b643

Non-examinable

Feature Reduction using PCA

$$X\tilde{e} = Z$$

num_samples x n

(n-dim features)

n x k

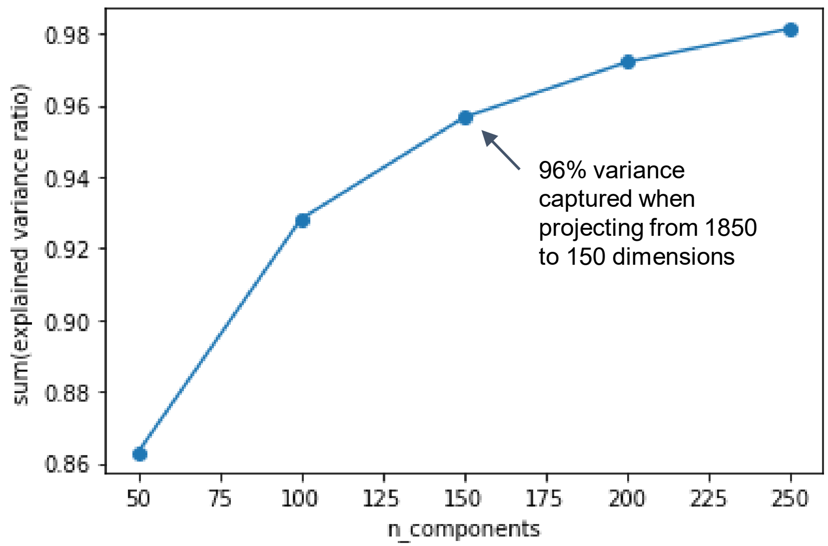
(k principal components)

num_samples x k

(k-dim projection)

PCA: Tuning

- Optimum dimension = maximize Explained Variance Ratio



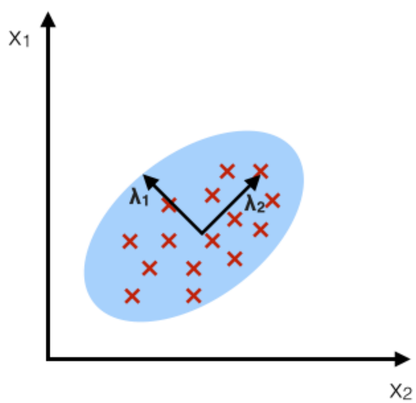
Explained variance = eigenvalue

Explained variance ratio = eigenvalue / sum(eigenvalues)

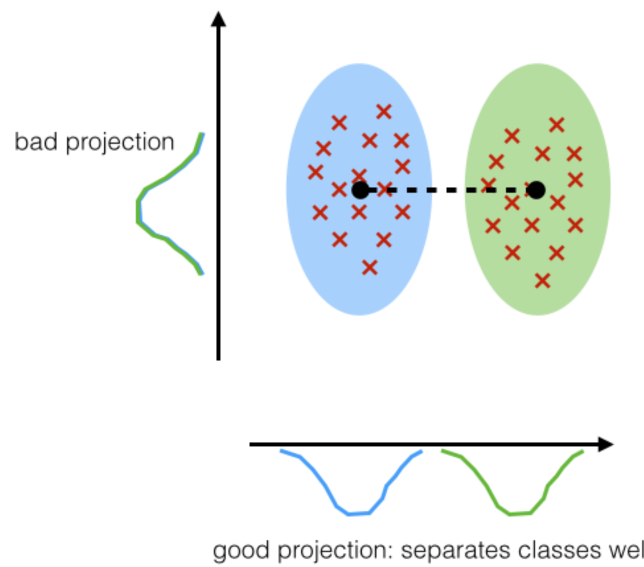
Somewhat similar to elbow method in k-means

Linear Discriminant Analysis (LDA)

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



LDA

Non-examinable

- Instead of Covariance Matrix, use Scatter Matrices $S_W^{-1} S_B$ to compute eigenvectors and eigenvalues

Within-class Scatter Matrix:

$$S_W = \sum_{i=1}^{c} S_i$$

$$S_i = \sum_{x \in D_i}^{n} (x - \mu_i)\,(x - \mu_i)^T$$

$$\mu_i = \frac{1}{n_i} \sum_{x \in D_i}^{n_i} x_k$$

Between-class Scatter Matrix:

$$S_B = \sum_{i=1}^{c} n_i (\mu_i - \mu)(\mu_i - \mu)^T$$

LDA: Intuition

$$S_W^{-1} S_B$$

Minimize within-class separation

Maximize between-class separation

Note: LDA is **supervised** - it uses the class labels. Only limited to **classification problems**

Reference: Linear Discriminant Analysis - bit by bit

Applications of Feature Reduction

- Feature reduction applies generally to every non-trivial dataset, but here are some interesting examples:
- Health: Gene Classification dataset
- Retail: Wine reviews dataset
- Transport: NYC taxi ride duration dataset
- Economic: Happiness and Employee Turnover dataset

Further study

- SHAP: SHapley Additive exPlanations
- Other Techniques
- t-SNE
- Select K best features (uses ANOVA)
- Manifold learning (e.g. Isomap)