

SPPU-BE-COMP-CONTENT – KSKA Git

ML UNIT 2 – PYQ Answers

➤ OCT 2022

Q3)

a) Consider a vector $x = (23, 29, 52, 31, 45, 19, 18, 27)$ Apply feature scaling and find out min-max scaled values as well as z-score values. [8]

➔ Don't know ! study yourself !

b) Explain the process of Principal Component Analysis (PCA) in brief [7]

Principal Component Analysis (PCA) – Process

Principal Component Analysis (PCA) is a technique used to reduce the dimensionality of large datasets, while retaining most of the variation present in the original data. This helps in simplifying data, removing noise, and improving computational efficiency.

- 1. Standardize the Data:**
Each feature is transformed to have zero mean and unit variance.
This step ensures all variables contribute equally and removes bias due to different scales.
- 2. Compute the Covariance Matrix:**
A covariance matrix is calculated to measure how variables vary together.
For a dataset with n features, this results in an $n \times n$ matrix representing relationships between features.
- 3. Calculate Eigenvalues and Eigenvectors:**
Eigenvectors represent the directions (principal components) of maximum variance in the data, while eigenvalues indicate the amount of variance captured by each principal component.
- 4. Sort and Select Principal Components:**
Eigenvalues are sorted in descending order.
The top k eigenvectors corresponding to the largest eigenvalues are selected as principal components.
These components explain most of the variance (commonly 90-95%).
- 5. Transform Original Data:**
The original data is projected onto the selected principal components, resulting in a reduced feature set with minimal loss of information.

Example:

If a dataset has 5 features, PCA might reduce it to 2 or 3 principal components that explain most of the variance. This makes visualization easier and speeds up further analysis.

PCA is a powerful tool to simplify complex datasets by extracting key features, reducing redundancy, and enhancing the performance of machine learning algorithms.

Q4)

a) How to handle missing values in a dataset that will be used for training the ML model? [5]

Handling Missing Values in a Dataset for Machine Learning

Missing data can negatively impact the performance of machine learning models. Common ways to handle missing values include:

1. Remove Missing Data

- Remove rows or columns with missing values if they are few or not significant.
- Suitable when missingness is random and the data loss is minimal.

2. Imputation Methods

- **Mean/Median Imputation:** Replace missing values with the mean or median of the feature column. Median is preferred for skewed data.
- **Mode Imputation:** For categorical features, replace missing values with the most frequent category.
- **Predictive Imputation:** Use regression or other ML models to predict missing values based on other features.

3. Use Algorithms That Handle Missing Data

- Some algorithms like decision trees can handle missing values internally.

4. Flag Missing Values

- Create an additional binary feature indicating whether a value was missing, helping the model learn patterns related to missingness.

5. Advanced Techniques

- Use methods like K-Nearest Neighbors (KNN) imputation or Multiple Imputation for better accuracy.

b) Explain the types of wrapper methods for feature selection. [5]

Wrapper methods select features based on the performance of a specific machine learning algorithm. They evaluate different subsets of features by training the model multiple times, using the model's accuracy (or other metrics) as a criterion to choose the best subset. This approach treats the model as a "black box" and searches for the optimal feature combination that yields the best predictive performance.

The main types of wrapper methods are:

1. Forward Selection:

- This method starts with an empty feature set (no features selected).

SPPU-BE-COMP-CONTENT – KSKA Git

- It adds one feature at a time by testing each remaining feature and selecting the one that improves model performance the most.
- The process repeats, adding features one by one until no further improvement is observed or a stopping criterion is met.
- Forward selection is useful when you want to build the feature set incrementally, ensuring only useful features are included.

2. **Backward Elimination:**

- This starts with all available features included in the model.
- At each step, it removes the least important feature whose removal either improves or minimally decreases the model's performance.
- The process continues until removing more features harms the model's accuracy or a predefined number of features remains.
- Backward elimination is good when you want to reduce a large feature set to the most relevant ones by pruning unnecessary features.

3. **Recursive Feature Elimination (RFE):**

- RFE uses the model itself to rank feature importance, often based on coefficients or feature importance scores (e.g., in decision trees).
- It recursively removes the least important feature(s) and retrains the model on the remaining features.
- This process repeats until the desired number of features is left.
- RFE combines the ideas of backward elimination and ranking to efficiently identify the most relevant features.

c) **Explain Local Binary Pattern (LBP) feature extraction technique with suitable example. [5]**

Local Binary Pattern (LBP) is a simple yet powerful texture descriptor used in image processing and computer vision. It captures the local spatial patterns and texture of an image by comparing each pixel with its neighbors.

How LBP Works:

1. Select a Pixel and Its Neighborhood:

- For each pixel in the image (called the center pixel), consider its surrounding neighbors (usually 8 neighbors in a 3×3 window).

2. Compare Neighboring Pixels:

- Compare each neighbor's intensity value with the center pixel's value.

SPPU-BE-COMP-CONTENT – KSKA Git

- Assign 1 if the neighbor's value is greater than or equal to the center pixel; otherwise, assign 0.
- 3. **Form a Binary Pattern:**
 - The resulting 0s and 1s form an 8-bit binary number (starting from a fixed neighbor and moving clockwise).
- 4. **Calculate the LBP Value:**
 - Convert the binary pattern to a decimal value.
 - This decimal number is the LBP value of the center pixel, representing the local texture pattern.
- 5. **Create LBP Histogram:**
 - Repeat the process for every pixel to get an LBP image.
 - Compute a histogram of LBP values over the image or regions to describe the texture.

Example:

Consider a 3×3 pixel patch:

50	55	45
60	52	48
58	53	47

- Center pixel value = 52
- Compare neighbors with 52:

50 (<52 →0)	55 (≥52 →1)	45 (<52 →0)
60 (≥52 →1)	52	48 (<52 →0)
58 (≥52 →1)	53 (≥52 →1)	47 (<52 →0)

- Binary pattern (starting top-left, clockwise): 0 1 0 0 0 1 1 1
- Binary to decimal: $0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 0 + 64 + 0 + 0 + 0 + 4 + 2 + 1 = 71$
- LBP value for the center pixel = 71

... "उत्तर बरोबर आहे का माहित नाही, एकदा चेक करा."

➤ SEP 2023

Q3)

a) What is feature selection? Explain filtering technique [5]

Feature Selection

Feature selection is the process of choosing a subset of relevant features (variables) from the original dataset to improve model performance, reduce overfitting, decrease training time, and simplify the model. It helps in removing redundant, irrelevant, or noisy features that do not contribute meaningfully to the prediction task.

Example:

In a dataset with 50 features, feature selection might identify that only 10 features are most important for predicting the target variable, improving accuracy and interpretability.

Filtering Technique

Filtering is a feature selection method that evaluates each feature independently based on statistical measures, without involving any machine learning algorithm. Features are ranked according to a scoring criterion, and those with scores above a threshold are selected.

Common Filtering Methods:

- **Correlation Coefficient:** Measures linear relationship between each feature and the target. Features with high correlation are selected.
- **Chi-Square Test:** Used for categorical data to test independence between feature and target.
- **Mutual Information:** Measures the amount of shared information between feature and target, capturing non-linear relationships.

Advantages:

- Fast and simple to compute.
- Works well for high-dimensional data.

Limitations:

- Ignores feature interactions and dependencies.
- Might select redundant features.

b) Explain kernel PCA in detail. [5]

Kernel Principal Component Analysis (Kernel PCA)

SPPU-BE-COMP-CONTENT – KSKA Git

Kernel PCA is an extension of the standard Principal Component Analysis (PCA) that allows for nonlinear dimensionality reduction by using kernel methods. Unlike linear PCA, which can only capture linear relationships, kernel PCA can extract nonlinear patterns from data.

How Kernel PCA Works:

1. Nonlinear Mapping:

- Data points from the original input space are mapped into a higher-dimensional feature space using a nonlinear function ϕ (phi).
- This mapping helps to make the data linearly separable or more structured in the new space.

2. Kernel Trick:

- Instead of computing the mapping ϕ explicitly (which may be computationally expensive or infinite-dimensional), kernel PCA uses a kernel function $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$.
- Common kernels include Gaussian (RBF), Polynomial, and Sigmoid kernels.

3. Compute Kernel (Gram) Matrix:

- Calculate the kernel matrix \mathbf{K} for all pairs of data points in the dataset, representing their similarity in the feature space.

4. Center the Kernel Matrix:

- The kernel matrix is centered to have zero mean in the feature space, a necessary step for PCA.

5. Eigen Decomposition:

- Perform eigen decomposition on the centered kernel matrix to obtain eigenvalues and eigenvectors.
- The eigenvectors correspond to principal components in the transformed space.

6. Project Data:

- Project the original data onto the principal components (eigenvectors) to get the reduced-dimensional representation.

c) Calculate LBP code generated value for the central point in the neighborhood of 8 pixels as shown below.

10	12	18
7	9	6
9	2	4

➔ Don't know ! solve IT !

Q4)

a) Explain Min-Max scaling with suitable example. [5]

Min-Max scaling is a normalization technique used to rescale the values of a feature to a fixed range, usually between 0 and 1. It transforms the data linearly so that the minimum value becomes 0 and the maximum value becomes 1, preserving the relationships between the original data points.

Formula:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Where:

- X = original value
- X_{\min} = minimum value of the feature
- X_{\max} = maximum value of the feature
- X_{scaled} = scaled value between 0 and 1

Example:

Suppose we have a feature with values:

20, 30, 40, 50, 60

- Minimum (X_{\min}) = 20
- Maximum (X_{\max}) = 60

To scale the value 40:

$$X_{\text{scaled}} = \frac{40 - 20}{60 - 20} = \frac{20}{40} = 0.5$$

Similarly, value 30 will be scaled to:

$$\frac{30 - 20}{60 - 20} = \frac{10}{40} = 0.25$$

Advantages:

- Scales data to a fixed range, useful for algorithms sensitive to data scale (e.g., k-NN, neural networks).
- Preserves the shape of the original distribution.

Limitations:

- Sensitive to outliers, as they affect the minimum and maximum values.

b) What is Matrix factorization? Explain content based filtering with an example. [5]

Matrix Factorization

Matrix factorization is a technique (often used in recommendation systems) where we break down a large matrix into the **product of smaller matrices** to learn hidden relationships between rows and columns.

For example, in a **movie recommendation system**, you might have a **user–movie rating matrix**:

User/Movie	Movie A	Movie B	Movie C	Movie D
User 1	5	?	3	?
User 2	?	4	?	2
User 3	4	?	4	?

Here, **?** means the rating is missing (user hasn't watched the movie).

Matrix factorization decomposes this **users × movies** matrix into two smaller matrices:


- **User matrix** (User → latent factors, e.g., preference for comedy, action, romance...)
- **Movie matrix** (Movie → latent factors, e.g., how much comedy, action, romance each movie has)

When we multiply these smaller matrices back together, we **predict missing ratings**.

Content-Based Filtering

Content-based filtering is a recommendation technique that suggests items **similar to what the user liked before**, based on item features.

How it works

1. Describe each item by its **features** (e.g., genre, keywords, description).
2. Track what a user likes or interacts with.
3. Recommend **other items with similar features**. 

SPPU-BE-COMP-CONTENT – KSKA Git

How it works

1. Describe each item by its **features** (e.g., genre, keywords, description).
2. Track what a user likes or interacts with.
3. Recommend **other items with similar features**.

Example

Suppose we have this movie database:

Movie	Genre
Inception	Sci-Fi, Action
Interstellar	Sci-Fi, Drama
Titanic	Romance, Drama
Avengers	Action, Sci-Fi

If **User A** liked "Inception" (Sci-Fi, Action), the system finds movies with **similar features** — "Avengers" (Action, Sci-Fi) and "Interstellar" (Sci-Fi, Drama) — and recommends them.

Difference from Collaborative Filtering:

- Content-based uses **item properties** (features)
- Collaborative filtering uses **user-item interactions** (matrix factorization is one way to do this)

SPPU-BE-COMP-CONTENT – KSKA Git

c) Given following data for attribute AGE calculate Z- score normalization. AGE = {18, 22, 25, 42, 28, 43, 33, 35, 56, 28} [5]


First, we calculate the **mean** and **standard deviation** for AGE:


- Mean (μ) = 33.0
- Standard deviation (σ) \approx 10.8351

The **Z-score** formula is:

$$Z = \frac{X - \mu}{\sigma}$$

Applying this to each value:

AGE	Z-score	
18	-1.3844	
22	-1.0152	
25	-0.7383	
42	0.8306	
28	-0.4615	
43	0.9229	
33	0.0000	
35	0.1846	
56	2.1227	
28	-0.4615	

So each value is transformed into how many **standard deviations** it is from the mean. 



SPPU-BE-COMP-CONTENT – KSKA Git

➤ SEP 2024

Q3)

a) Convert given data set in normalized data set. $D = \{23, 29, 52, 31, 45, 19, 18, 27\}$. [8]

To normalize a dataset, we usually use Min-Max normalization to scale values between 0 and 1.

The formula is:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Given:

$D = \{23, 29, 52, 31, 45, 19, 18, 27\}$

- $X_{min} = 18$
- $X_{max} = 52$

Calculations:

$$X' = \frac{X - 18}{52 - 18} = \frac{X - 18}{34}$$

X	Normalized X'
23	0.1471
29	0.3235
52	1.0000
31	0.3824
45	0.7941
19	0.0294
18	0.0000
27	0.2647

b) Why do you need categorical variable encoding? With an example, explain one-hot encoding.

Categorical variable encoding is needed because most machine learning algorithms work with numerical inputs, not directly with text or category labels.

If your dataset has categorical data like "Red", "Blue", "Green", you can't feed these strings directly into models — they need to be converted into numbers in a way that preserves meaning.

If you assign numbers directly (e.g., Red = 1, Blue = 2, Green = 3), the model might mistakenly think there's an order or distance between the categories (as if Green > Blue), which is wrong for purely nominal data.

One-Hot Encoding

One-hot encoding solves this problem by creating new binary columns — one for each category — and marking a 1 in the column that matches the value, and 0 for the rest.

SPPU-BE-COMP-CONTENT – KSKA Git

Example

Suppose you have a dataset:

Color

Red

Blue

Green

Blue

After one-hot encoding:

Color_Red	Color_Blue	Color_Green
1	0	0
0	1	0
0	0	1
0	1	0

Why it's useful

- No false ordinal relationships between categories
- Works well for algorithms that can't handle raw strings (like Logistic Regression, SVMs, Neural Networks)
- Preserves full categorical meaning

Q4)

a) Which statistical methods are used to describe the nature of data? [5]

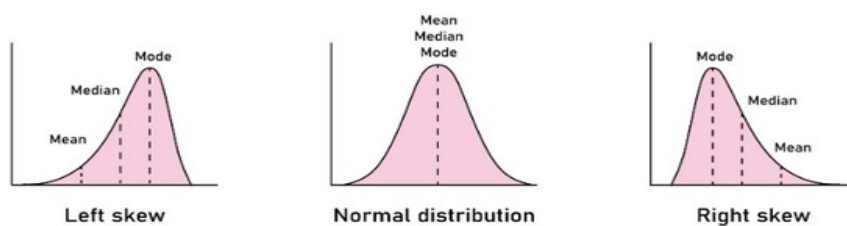
Statistical Methods to Describe the Nature of Data

Statistical methods help summarize and describe the main features of a dataset, giving insights into its distribution, central tendency, and variability. The common statistical methods include:

1. Measures of Central Tendency:

- **Mean:** Average value of the data points.
- **Median:** Middle value when data is ordered, useful for skewed data.
- **Mode:** Most frequently occurring value.

Mean, Median and Mode



2. Measures of Dispersion (Spread):

SPPU-BE-COMP-CONTENT – KSKA Git

- **Range:** Difference between the maximum and minimum values.
 - **Variance:** Average of squared differences from the mean, showing spread.
 - **Standard Deviation:** Square root of variance; indicates how much data varies from the mean.
3. **Measures of Shape:**
- **Skewness:** Indicates asymmetry of the data distribution (positive or negative skew).
 - **Kurtosis:** Measures the "tailedness" or peak sharpness of the distribution.
4. **Frequency Distribution:**
- Tabulates the number of occurrences of each value or range of values. Often visualized by histograms.
5. **Percentiles and Quartiles:**
- Percentiles divide data into 100 equal parts; quartiles divide into four parts, showing data spread and outliers.

These statistical methods collectively help in understanding the overall pattern, variability, and shape of the data, which is essential for further analysis or modeling.

b) What are the feature of multidimensional scaling? [5]

Features of Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS) is a dimensionality reduction technique used to visualize the similarity or dissimilarity between data points in a lower-dimensional space. Its key features include:

1. **Preserves Pairwise Distances:**
MDS aims to maintain the original distances (or dissimilarities) between data points as closely as possible in the reduced-dimensional space.
2. **Works with Any Distance Metric:**
MDS can use various types of distance or dissimilarity measures, such as Euclidean, Manhattan, or correlation-based distances, making it flexible for different data types.
3. **Unsupervised Technique:**
It does not require labeled data; it only uses the pairwise distance or similarity matrix as input.
4. **Produces a Geometric Representation:**
MDS provides a spatial map where points that are more similar are placed closer together, aiding in visual interpretation of complex relationships.
5. **Handles Nonlinear Relationships:**
Some variants of MDS (like non-metric MDS) can capture nonlinear structures in data, making it more powerful than linear methods like PCA for certain datasets.

c) Elaborate use of PCA in preprocessing stage. [5]

Use of PCA in the Preprocessing Stage

Principal Component Analysis (PCA) is widely used as a preprocessing step in machine learning and data analysis to improve model performance and efficiency. Its main uses in preprocessing include:

1. Dimensionality Reduction:

- PCA reduces the number of input features by transforming the original variables into a smaller set of uncorrelated variables called principal components.
- This helps to simplify the dataset without losing significant information, making the training process faster and less computationally expensive.

2. Noise Reduction:

- By focusing on components with the highest variance, PCA filters out components that mostly capture noise or insignificant variations, improving the signal-to-noise ratio.

3. Dealing with Multicollinearity:

- PCA transforms correlated features into a set of orthogonal (uncorrelated) components, addressing multicollinearity problems that can degrade the performance of some models.

4. Data Visualization:

- PCA helps in projecting high-dimensional data onto 2D or 3D space, making it easier to visualize patterns, clusters, or outliers in the data before applying machine learning algorithms.

5. Improving Model Generalization:

- Reducing dimensionality and noise often leads to better generalization on unseen data by preventing overfitting.

Example:

In an image recognition task, PCA can reduce thousands of pixel features to a few principal components that capture most of the important variance, making the model training more efficient without losing key information.

"Check/Verify Answer – Read at Your Own Risk"