Here's a complete answer key for **K-Means Clustering** and **Hierarchical Clustering** based on the questions you provided — fully explained with step-by-step examples and diagrams, optimized for 9-mark answers:

**✅ Q1. Explain steps of K-Means with example (2022)**

📌 **Answer (with example):**

**K-Means Clustering Steps:**

1. **Choose K** – Select number of clusters (K).
2. **Initialize Centroids** – Randomly choose K points as initial centroids.
3. **Assign Points to Nearest Centroid** – Use distance formula (usually Euclidean).
4. **Update Centroids** – Compute mean of all points assigned to each cluster.
5. **Repeat Steps 3 & 4** until centroids do not change (convergence).

📘 **Example:**

Dataset:

Points: A(2,3), B(4,7), C(3,5), D(6,9), E(8,6), F(7,8)

Initial Centroids: C1(2,3), C2(8,6)

| **Point** | **Distance to C1 (2,3)** | **Distance to C2 (8,6)** | **Assigned Cluster** |
| --- | --- | --- | --- |
| A | 0 | 6.7 | C1 |
| B | 5.4 | 4.1 | C2 |
| C | 2.2 | 5.0 | C1 |
| D | 7.2 | 3.2 | C2 |
| E | 6.0 | 0.0 | C2 |
| F | 6.4 | 2.2 | C2 |

🔁 Update Centroids

* **New C1**: Mean of A(2,3) and C(3,5) → ((2+3)/2, (3+5)/2) = (2.5, 4)
* **New C2**: Mean of B, D, E, F → ((4+6+8+7)/4, (7+9+6+8)/4) = (6.25, 7.5)

✍️ Repeat assignment with new centroids until centroids stop changing.

📈 **Diagram** (ASCII style):

C1 o A

C

C1'

B F D

|

E C2'

**✅ Q2. Cluster the given points using K-means. Show first iteration. (2023 Jan, 2023 Dec)**

Use same dataset:

Points: A(2,3), B(4,7), C(3,5), D(6,9), E(8,6), F(7,8)

Initial Centroids: C1(2,3), C2(8,6)

**Step 1: Distance Calculation**

| **Point** | **Distance to C1 (2,3)** | **Distance to C2 (8,6)** | **Assigned Cluster** |
| --- | --- | --- | --- |
| A | 0 | 6.7 | C1 |
| B | √((4−2)²+(7−3)²)=√20 ≈ 4.47 | √((4−8)²+(7−6)²)=√17 ≈ 4.12 | C2 |
| C | √((3−2)²+(5−3)²)=√5 ≈ 2.24 | √((3−8)²+(5−6)²)=√26 ≈ 5.10 | C1 |
| D | √((6−2)²+(9−3)²)=√52 ≈ 7.21 | √((6−8)²+(9−6)²)=√13 ≈ 3.61 | C2 |
| E | √((8−2)²+(6−3)²)=√45 ≈ 6.71 | 0 | C2 |
| F | √((7−2)²+(8−3)²)=√50 ≈ 7.07 | √((7−8)²+(8−6)²)=√5 ≈ 2.24 | C2 |

✅ First iteration result:

* **Cluster 1**: A, C
* **Cluster 2**: B, D, E, F

**✅ Q3. Perform full K-means clustering until convergence (2024 May)**

**Points: A(2,3), B(4,7), C(3,5), D(6,9), E(8,6), F(7,8)**  
Initial Centroids: C1(2,3), C2(8,6), K=2

**🌀 Iteration 1**

Same as above.  
Clusters:

* C1: A, C
* C2: B, D, E, F

New Centroids:

* **C1** = (2.5, 4)
* **C2** = (6.25, 7.5)

**🌀 Iteration 2**

Reassign points using new centroids.

| **Point** | **Dist to C1(2.5,4)** | **Dist to C2(6.25,7.5)** | **Assigned** |
| --- | --- | --- | --- |
| A | √((2−2.5)²+(3−4)²)=√1.25 ≈ 1.12 | √29.25 ≈ 5.4 | C1 |
| B | √((4−2.5)²+(7−4)²)=√11.25 ≈ 3.35 | √(2.25+0.25)=√2.5 ≈ 1.58 | C2 |
| C | √((3−2.5)²+(5−4)²)=√1.25 ≈ 1.12 | √(10.56+0.25)=√10.81 ≈ 3.29 | C1 |
| D | √(12.25+25)=√37.25 ≈ 6.10 | √(−0.25)² + (1.5)² = √0.0625+2.25 = √2.31 ≈ 1.52 | C2 |
| E | √(30.25+4)=√34.25 ≈ 5.85 | √((8−6.25)²+(6−7.5)²) = √3.06+2.25 = √5.31 ≈ 2.3 | C2 |
| F | √((7−2.5)²+(8−4)²)=√20.25+16 = √36.25 ≈ 6.02 | √((7−6.25)²+(8−7.5)²)=√0.5625 ≈ 0.75 | C2 |

No change in clusters → **Converged**

**Final Clusters**:

* Cluster 1: A, C
* Cluster 2: B, D, E, F
* Final Centroids: (2.5, 4), (6.25, 7.5)

Here is the **detailed answer** with each question part mentioned exactly as in the image:

**Q5 a) K-means Algorithm – First Round of Execution**

**Question:**  
Suppose that the given data the task is to cluster points (With (x,y) representing location) into three clusters, where the points are:  
A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4), C1(1,2), C2(4,9)  
The distance function is Euclidean distance. Suppose initially we assign A1, B1, and C1 as the center of each cluster, respectively. Use the k-means algorithm to show only the three cluster centers after the first round of execution with steps.

**Step 1: Initial Cluster Centers**

* Cluster 1: A1(2,10)
* Cluster 2: B1(5,8)
* Cluster 3: C1(1,2)

We will now assign each point to the nearest cluster center using **Euclidean distance**:

Distance=(x2−x1)2+(y2−y1)2\text{Distance} = \sqrt{(x\_2 - x\_1)^2 + (y\_2 - y\_1)^2}

**Step 2: Distance Calculations and Assignments**

**A2(2,5)**

* To A1: √((2−2)² + (5−10)²) = √25 = 5
* To B1: √((2−5)² + (5−8)²) = √18 ≈ 4.24
* To C1: √((2−1)² + (5−2)²) = √10 ≈ 3.16 → **Cluster 3**

**A3(8,4)**

* A1: √((8−2)² + (4−10)²) = √72 ≈ 8.49
* B1: √((8−5)² + (4−8)²) = √25 = 5
* C1: √((8−1)² + (4−2)²) = √53 ≈ 7.28 → **Cluster 2**

**B2(7,5)**

* A1: √((7−2)² + (5−10)²) = √50 ≈ 7.07
* B1: √((7−5)² + (5−8)²) = √13 ≈ 3.61
* C1: √((7−1)² + (5−2)²) = √45 ≈ 6.70 → **Cluster 2**

**B3(6,4)**

* A1: √((6−2)² + (4−10)²) = √52 ≈ 7.21
* B1: √((6−5)² + (4−8)²) = √17 ≈ 4.12
* C1: √((6−1)² + (4−2)²) = √29 ≈ 5.38 → **Cluster 2**

**C2(4,9)**

* A1: √((4−2)² + (9−10)²) = √5 ≈ 2.24
* B1: √((4−5)² + (9−8)²) = √2 ≈ 1.41
* C1: √((4−1)² + (9−2)²) = √58 ≈ 7.62 → **Cluster 2**

**Summary of Assignments:**

* **Cluster 1 (A1):** Only A1
* **Cluster 2 (B1):** A3, B2, B3, C2
* **Cluster 3 (C1):** A2, C1

**Step 3: New Cluster Centers**

**Cluster 1 (A1):**

Only A1(2,10) → Center = (2,10)

**Cluster 2:**

Points: A3(8,4), B2(7,5), B3(6,4), C2(4,9)

* Avg x = (8+7+6+4)/4 = 6.25
* Avg y = (4+5+4+9)/4 = 5.5  
  → New center: **(6.25, 5.5)**

**Cluster 3:**

Points: A2(2,5), C1(1,2)

* Avg x = (2+1)/2 = 1.5
* Avg y = (5+2)/2 = 3.5  
  → New center: **(1.5, 3.5)**

**Final Answer (After First Round):**

* **Cluster 1 center:** (2, 10)
* **Cluster 2 center:** (6.25, 5.5)
* **Cluster 3 center:** (1.5, 3.5)

**✅ Q4. Explain how hierarchical clustering can be used for visualizing hierarchical relationships in data. Give example and applications.**

📌 **Answer:**

**Hierarchical Clustering**:  
A method to build a hierarchy of clusters either:

* **Agglomerative (bottom-up)** – start with individual points and merge clusters
* **Divisive (top-down)** – start with one cluster and split

🔍 **Dendrogram**:  
A tree-like diagram used to visualize cluster merging steps.

📘 **Example**:

| **Points** | **Distances** |
| --- | --- |
| A–B | 2 |
| A–C | 4 |
| B–C | 3 |

Agglomerative Steps:

1. Merge A and B → AB
2. Merge AB with C → ABC

🖼 **Dendrogram (ASCII)**:

\_\_\_\_\_\_\_\_\_

| |

AB C

/ \

A B

🎯 **Applications**:

* **Gene Expression Analysis**
* **Market Segmentation**
* **Document Clustering**
* **Image Segmentation**
* **Social Network Analysis**

Let me know if you'd like **handwritten-style notes or image diagrams** for these!

Sure! Here’s a detailed **9-mark answer** on **Hierarchical Clustering Algorithms**, covering types, steps, advantages, and key methods:

**Hierarchical Clustering Algorithms (9 Marks)**

Hierarchical clustering builds a hierarchy of clusters, represented by a tree called a **dendrogram**. It is mainly of two types:

**1. Agglomerative Hierarchical Clustering (Bottom-Up Approach)**

* Start with **each data point as a single cluster** (total N clusters for N points).
* At each step, **merge the two closest clusters** based on a chosen distance metric.
* Repeat merging until **all points form one single cluster** or the desired number of clusters is reached.

**Key Steps:**

1. Compute the **distance matrix** between all data points.
2. Find the **closest two clusters** (minimum distance).
3. **Merge** these two clusters.
4. Update the distance matrix to reflect the new cluster.
5. Repeat steps 2–4 until stopping criteria.

**Linkage Methods** (to measure distance between clusters):

* **Single linkage**: Distance between nearest points of clusters
* **Complete linkage**: Distance between farthest points
* **Average linkage**: Average distance between all points in clusters
* **Ward’s method**: Minimizes variance within clusters

**Agglomerative Clustering Example and Dendrogram**

Given points: A, B, C, D

| **Step** | **Clusters Merged** | **Distance** |
| --- | --- | --- |
| 1 | A and B | 2 |
| 2 | C and D | 3 |
| 3 | (A,B) and (C,D) | 5 |

Distance

5 | \_\_\_\_\_\_\_\_\_

| | |

3 | \_\_\_|\_\_\_ |

| | | |

2 | A B C---D

|

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A B C D

**2. Divisive Hierarchical Clustering (Top-Down Approach)**

* Start with **all data points in one cluster**.
* At each step, **split a cluster** into smaller clusters.
* Repeat until each point is a single cluster or desired clusters formed.

**Less commonly used** due to complexity.

**Distance Metrics**

* **Euclidean distance**
* **Manhattan distance**
* **Cosine similarity** (for text data)

**Visualization – Dendrogram**

* Shows the sequence of merges/splits.
* Y-axis indicates distance or dissimilarity.
* Cutting dendrogram at a certain height gives the desired number of clusters.

**Advantages**

* No need to pre-specify number of clusters (can decide by cutting dendrogram).
* Easy to visualize data hierarchy.
* Works well for small datasets.

**Disadvantages**

* Computationally expensive for large datasets (O(n³)).
* Sensitive to noise and outliers.

**Example Application**

* **Gene expression data analysis** to find related genes.
* **Document clustering** in text mining.

**Summary Table:**

| **Algorithm Type** | **Description** | **Steps** | **Linkage Methods** | **Use Case** |
| --- | --- | --- | --- | --- |
| Agglomerative | Merge closest clusters iteratively | Start with points as clusters, merge until one cluster | Single, Complete, Average, Ward | Small-medium data clustering |
| Divisive | Split clusters iteratively | Start with all points, split until singletons | N/A | Rarely used, complex datasets |

If you want, I can also prepare a **simple dendrogram diagram** or **stepwise example** for this!

**validation techniques**

Sure! Here are detailed 9-mark answers for each of your questions on, including explanations, comparisons, and diagrams where useful:

**1. Discuss Holdout and Random Subsampling (9 Marks)**

**Holdout Method:**

* The dataset is split **once** into two (or three) mutually exclusive sets:
  + **Training set:** Used to train the model (typically 60-80% of data).
  + **Test set:** Used to evaluate the model's performance (remaining 20-40%).
* Simple and fast, but results depend heavily on the split, which may be non-representative.
* Common in large datasets where repeated validation is expensive.

**Random Subsampling (Repeated Holdout):**

* The dataset is randomly split into training and test sets **multiple times**.
* Each time, a model is trained and evaluated.
* Results are averaged to get more reliable performance estimates.
* Helps reduce bias and variance compared to a single holdout split.

**Comparison:**

| **Aspect** | **Holdout** | **Random Subsampling** |
| --- | --- | --- |
| Number of splits | 1 | Multiple (repeated) |
| Bias | High (depends on split) | Lower (averages over splits) |
| Computational cost | Low | Higher |
| Variance | High | Lower |
| Use case | Large datasets, quick eval | Smaller datasets, more reliable |

**Diagram for Holdout and Random Subsampling:**

Holdout:

Data --------------------> Train (70%) ------> Model Train

\

--> Test (30%) ------> Model Evaluate

Random Subsampling:

Repeat N times:

Data -> Random Split -> Train -> Model Train

\

-> Test -> Model Evaluate

Average performance over N runs.

**2. Explain k-fold Cross Validation & Random Subsampling (9 Marks)**

**k-fold Cross Validation:**

* Dataset is divided into **k equal-sized folds** (typically k=5 or 10).
* The model is trained k times, each time:
  + Using k-1 folds as training data.
  + Using the remaining 1 fold as validation data.
* Performance metrics are averaged over k trials.
* Reduces bias and variance in performance estimate.
* Common and reliable for small to medium datasets.

**Steps of k-fold CV:**

1. Split data into k folds.
2. For i = 1 to k:
   * Train on all folds except fold i.
   * Validate on fold i.
3. Average performance metrics.

**Random Subsampling:**

* Explained above, randomly split multiple times.
* No guarantee all data used for training/validation exactly once.
* Slightly less exhaustive than k-fold CV.
* Useful for large datasets.

**Comparison:**

| **Aspect** | **k-fold CV** | **Random Subsampling** |
| --- | --- | --- |
| Number of splits | Exactly k | Variable, chosen by user |
| Data usage | Every point used exactly once for validation | Random subsets, may overlap |
| Bias | Low | Low to moderate |
| Variance | Moderate | Moderate to low |
| Computational cost | High | Medium to high |

**Diagram of 5-Fold CV:**

Data folds: [F1][F2][F3][F4][F5]

Iteration 1: Train on F2-F5, Test on F1

Iteration 2: Train on F1,F3-F5, Test on F2

Iteration 3: Train on F1,F2,F4,F5, Test on F3

Iteration 4: Train on F1-F3,F5, Test on F4

Iteration 5: Train on F1-F4, Test on F5

**3. Holdout Method and Train/Validate/Test Split (9 Marks)**

**Holdout Method:**

* Split dataset into **training** and **testing** sets.
* Train model on training data.
* Evaluate on testing data.
* Simple but risk of overfitting or underfitting due to single split.

**Train/Validate/Test Split:**

* Dataset split into **three subsets**:
  + **Training set:** Used to fit the model.
  + **Validation set:** Used for hyperparameter tuning and model selection.
  + **Test set:** Final evaluation on unseen data.
* Typical splits: 60% train, 20% validation, 20% test.
* Helps avoid overfitting on test data and gives unbiased performance estimate.

**Steps:**

1. Split data into Train/Validate/Test.
2. Train on training set.
3. Use validation set to tune hyperparameters and select model.
4. Once best model is chosen, evaluate final performance on test set.

**Advantages:**

* Clear separation for tuning and final evaluation.
* Prevents information leakage from test to training.

**Disadvantages:**

* Less data available for training.
* Results depend on how the split is done.

**Diagram for Train/Validate/Test Split:**

Data

|---------------------------|

| Train (60%) | Validate (20%) | Test (20%) |

Training --> Model Train

Validation --> Hyperparameter Tuning

Test --> Final Model Evaluation

If you want, I can also create a combined summary table or flowcharts for these methods!

**Text Analysis**

Here are detailed **9-mark answers** for each of your text analysis and social media network analysis questions, with explanations and examples:

**1. What is Text Analysis and Steps Involved in It? (9 Marks)**

**Text Analysis** (also called Text Mining or Natural Language Processing) is the process of extracting meaningful information from unstructured text data. It transforms raw text into structured data for analysis.

**Steps involved in Text Analysis:**

1. **Text Collection:**  
   Gather raw text data from sources like documents, social media, websites, emails.
2. **Text Preprocessing:**
   * **Tokenization:** Split text into words or sentences.
   * **Stop word removal:** Remove common words like "the," "is," which carry less meaning.
   * **Lowercasing:** Convert all text to lower case.
   * **Punctuation removal:** Remove commas, periods, etc.
3. **Normalization:**
   * **Stemming:** Reduce words to their root form (e.g., "running" → "run").
   * **Lemmatization:** Convert words to dictionary base form considering context (e.g., "better" → "good").
4. **Feature Extraction:**  
   Convert text into numeric features, e.g.,
   * Bag of Words (BoW)
   * TF-IDF (Term Frequency-Inverse Document Frequency)
5. **Text Representation:**  
   Represent text as vectors or matrices suitable for ML models.
6. **Text Analysis / Modeling:**  
   Apply classification, clustering, sentiment analysis, topic modeling.
7. **Interpretation & Visualization:**  
   Visualize results using word clouds, frequency plots, network graphs.

**Example:**

Analyzing customer reviews to extract sentiment using TF-IDF features and classify as positive or negative.

**2. What is Social Media Network Analysis and Its Applications? (9 Marks)**

**Social Media Network Analysis (SMNA)** is the study of social relationships and interactions on social media platforms by analyzing the network structure formed by users and their connections (e.g., followers, likes, shares).

**Key Concepts:**

* **Nodes:** Users or accounts.
* **Edges:** Relationships or interactions (friendship, following, retweets).
* **Network metrics:** Centrality (importance of node), clusters (communities), influence.

**Applications of SMNA:**

* **Influencer Identification:** Find key opinion leaders.
* **Community Detection:** Identify groups with common interests.
* **Sentiment Analysis:** Track public opinion on brands or events.
* **Information Diffusion:** Study how information spreads.
* **Marketing Campaigns:** Target influential users for promotions.
* **Crisis Management:** Detect and respond to viral negative events.

**Example:**

Analyzing Twitter retweet network to find most influential users during an election.

**3. Explain TF-IDF with Example (9 Marks)**

**TF-IDF (Term Frequency - Inverse Document Frequency)** is a statistical measure to evaluate how important a word is in a document relative to a collection (corpus).

* **Term Frequency (TF):** Number of times a term appears in a document / Total terms in document.
* **Inverse Document Frequency (IDF):**  
  IDF=log⁡(Total number of documentsNumber of documents containing the term)\text{IDF} = \log \left(\frac{\text{Total number of documents}}{\text{Number of documents containing the term}} \right)
* **TF-IDF Score:** TF×IDF\text{TF} \times \text{IDF}

**Example:**

Suppose we have 3 documents:

* Doc1: "Data science is fun"
* Doc2: "Machine learning and data science"
* Doc3: "Deep learning in AI"

Calculate TF-IDF of "data" in Doc2:

* TF = 1/4 (data appears once in 4 words)
* IDF = log⁡(3/2)=0.176\log(3/2) = 0.176 (since "data" appears in 2 documents)
* TF-IDF = 0.25×0.176=0.0440.25 \times 0.176 = 0.044

"data" is moderately important in Doc2.

**4. Explain POS Tagging, Lemmatization, Stemming (9 Marks)**

**POS Tagging (Part-of-Speech Tagging):**  
Assigns word classes (noun, verb, adjective, etc.) to each word in a sentence.  
Example: "I am reading a book." → I/PRON, am/VERB, reading/VERB, a/DET, book/NOUN

**Stemming:**  
A crude heuristic process that chops off word endings to get root form.  
Example: "running", "runner" → "run"  
It may produce non-dictionary words.

**Lemmatization:**  
More sophisticated process using vocabulary and morphological analysis to return the base/dictionary form (lemma) of a word.  
Example: "better" → "good", "running" → "run"

**Comparison Table:**

| **Technique** | **Purpose** | **Output Example** | **Notes** |
| --- | --- | --- | --- |
| POS Tagging | Label words by POS | run/VERB, book/NOUN | Context-dependent |
| Stemming | Reduce to root/stem | "running" → "run" | May create non-words |
| Lemmatization | Base dictionary form | "better" → "good" | Requires dictionary lookup |

**5. Explain TF-IDF and Bag of Words (9 Marks)**

**Bag of Words (BoW):**

* Represents text as the **frequency count** of words ignoring grammar and order.
* Creates a vector where each dimension corresponds to a vocabulary word.
* Simple, effective but treats all words equally, ignoring importance.

**TF-IDF:**

* Extends BoW by weighing words based on their importance.
* Rare words across documents get higher weight.
* Reduces the impact of common words.

**Example:**

For two documents:

* Doc1: "cat sat on mat"
* Doc2: "dog sat on log"

**BoW vectors:**

| **Word** | **Doc1 Count** | **Doc2 Count** |
| --- | --- | --- |
| cat | 1 | 0 |
| sat | 1 | 1 |
| on | 1 | 1 |
| mat | 1 | 0 |
| dog | 0 | 1 |
| log | 0 | 1 |

**TF-IDF** will reduce weights of "sat" and "on" as they appear in both docs, highlighting unique words like "cat" or "dog".

If you want, I can prepare diagrams or examples illustrating TF-IDF calculations or POS tagging!

**CONFUSION MATRIX**

**1. Confusion Matrix Terms: Accuracy, Precision, Recall, AUC-ROC (9 Marks)**

**Confusion Matrix** is a table summarizing the performance of a classification model by comparing predicted and actual classes. For binary classification, it has four values:

|  | **Predicted Positive** | **Predicted Negative** |
| --- | --- | --- |
| Actual Positive | True Positive (TP) | False Negative (FN) |
| Actual Negative | False Positive (FP) | True Negative (TN) |

**Metrics:**

1. **Accuracy:**
   * Measures overall correctness.

Accuracy=TP+TNTP+TN+FP+FN\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}

1. **Precision:**
   * How many predicted positives are actually positive.

Precision=TPTP+FP\text{Precision} = \frac{TP}{TP + FP}

1. **Recall (Sensitivity or True Positive Rate):**
   * How many actual positives are correctly predicted.

Recall=TPTP+FN\text{Recall} = \frac{TP}{TP + FN}

1. **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):**
   * ROC curve plots True Positive Rate (Recall) vs False Positive Rate (FPR = FPFP+TN\frac{FP}{FP + TN}) at various thresholds.
   * AUC-ROC summarizes model’s ability to discriminate positive and negative classes across thresholds.
   * Value ranges from 0.5 (random guessing) to 1 (perfect classification).
   * Higher AUC indicates better performance.

**Summary Table:**

| **Metric** | **Formula** | **Interpretation** |
| --- | --- | --- |
| Accuracy | (TP + TN) / Total | Overall correctness |
| Precision | TP / (TP + FP) | Correctness of positive predictions |
| Recall | TP / (TP + FN) | Coverage of actual positives |
| AUC-ROC | Area under TPR vs FPR curve | Discrimination ability of model |

**Q6 a) Confusion Matrix Calculations and Description**

**Question:**  
Given the confusion matrix, calculate accuracy, precision, recall, error rate with description on heart attack risk.

**Confusion Matrix:**

|  | **Predicted: Yes** | **Predicted: No** |
| --- | --- | --- |
| **Actual: Yes** | 80 (TP) | 220 (FN) |
| **Actual: No** | 150 (FP) | 9500 (TN) |

**Metrics Calculations:**

1. **Accuracy:**

=TP+TNTotal=80+950080+220+150+9500=95809950≈0.9628 or 96.28%= \frac{TP + TN}{Total} = \frac{80 + 9500}{80 + 220 + 150 + 9500} = \frac{9580}{9950} ≈ 0.9628 \text{ or } 96.28\%

1. **Precision:**

=TPTP+FP=8080+150=80230≈0.3478 or 34.78%= \frac{TP}{TP + FP} = \frac{80}{80 + 150} = \frac{80}{230} ≈ 0.3478 \text{ or } 34.78\%

1. **Recall (Sensitivity):**

=TPTP+FN=8080+220=80300≈0.2667 or 26.67%= \frac{TP}{TP + FN} = \frac{80}{80 + 220} = \frac{80}{300} ≈ 0.2667 \text{ or } 26.67\%

1. **Error Rate:**

=1−Accuracy=1−0.9628=0.0372 or 3.72%= 1 - \text{Accuracy} = 1 - 0.9628 = 0.0372 \text{ or } 3.72\%

**Description:**

The model is **highly accurate overall (96.28%)**, which means it correctly predicts most cases.  
However, the **precision (34.78%)** and **recall (26.67%)** for predicting heart attack risk are low. This indicates that:

* It predicts many non-risk individuals as at-risk (low precision).
* It fails to identify many truly at-risk individuals (low recall).

**Conclusion:** In a medical context like heart attack risk, **recall is crucial**, as missing actual at-risk patients can have serious consequences.

Let me know if you want this as a formatted PDF or Word file.