# MZ02 Milestones Compliance Report III: Finalized Design and Analysis

## Part 1: Infrastructure Design and Cost Analysis (Student C)

### 1. Finalized System Infrastructure Design

This section presents the finalized infrastructure design, transitioning from the high-level architectural concepts established in the previous reporting period to a detailed, operational specification. The following subsections detail the precise interaction patterns between the system's microservices and the specific configuration of the underlying Google Cloud Platform (GCP) services chosen for deployment.

#### 1.1 Microservices Interaction Model: API Call Flow

The system's functionality is delivered through a coordinated sequence of interactions between its independent microservices. A typical user query—a clinician checking for a potential interaction between two drugs—initiates a cascade of internal API calls orchestrated to deliver a comprehensive risk assessment.

The process begins when a clinician enters two drug names into the frontend dashboard and submits the query. This client-side action triggers a POST request to a secure endpoint on the API Gateway, for instance, /api/ddi-check.1 The API Gateway, serving as the single entry point, authenticates the request and routes it to the central Orchestration Service.

The Orchestration Service, designated as "The Brain" of the system, receives the request containing the drug names.1 It then executes the core business logic by coordinating calls to the various backend microservices. Its first action is to invoke the NLP Inference Service via a gRPC call, passing the drug names and any relevant context. The NLP Inference Service, which hosts the fine-tuned PubMedBERT model on a dedicated Vertex AI Endpoint, processes the input and returns a probability distribution over potential interaction types.1

Concurrently or sequentially, the Orchestrator queries the Database Service. This service, backed by a Cloud SQL instance, holds structured data from sources like DrugBank and RxNorm.1 The query serves to resolve drug synonyms (e.g., mapping "Tylenol" to "acetaminophen") and retrieve known interaction data, providing a layer of validation against established knowledge bases. The Orchestrator may also query an External Literature Service to fetch recent abstracts or articles relevant to the drug pair, providing clinicians with direct access to supporting evidence.

Once the Orchestrator has received responses from the NLP and Database services, it makes a final API call to the dedicated **Risk Scoring Service**. It passes the raw probabilities from the NLP model and any relevant context (such as known interaction data from the database) to this service. The Risk Scoring Service, which contains the algorithm logic developed by Student D, calculates the final risk score and categorical level, returning it to the Orchestrator. The Orchestrator then aggregates this definitive risk assessment, the structured database information, and links to external literature into a single, structured JSON response. This response is sent back through the API Gateway to the clinician's dashboard, where the results are rendered in a user-friendly format.

The following sequence diagram provides a formal, visual representation of this API call flow.

Code snippet

sequenceDiagram  
 participant Client as Clinician Dashboard  
 participant Gateway as API Gateway  
 participant Orchestrator as Orchestration Service  
 participant NLP as NLP Inference Service  
 participant DB as Database Service  
 participant ExtLit as External Literature Service  
  
 Client->>+Gateway: POST /api/ddi-check (JSON: {drug1, drug2})  
 Gateway->>+Orchestrator: Forward Request  
 Orchestrator->>+NLP: gRPC: process\_interaction({drug1, drug2})  
 NLP-->>-Orchestrator: Response (Probabilities)  
 Orchestrator->>+DB: SQL: query\_drugs({drug1, drug2})  
 DB-->>-Orchestrator: Response (Synonyms, Known Interactions)  
 Orchestrator->>+ExtLit: API: search\_literature({drug1, drug2})  
 ExtLit-->>-Orchestrator: Response (Article Links)  
 Orchestrator->>Orchestrator: Synthesize Data & Calculate Risk Score  
 Orchestrator-->>-Gateway: Aggregated JSON Response  
 Gateway-->>-Client: 200 OK (Risk Assessment)

A critical consideration in this architecture is the central role of the Orchestration Service. While this design simplifies the logic of other microservices by centralizing control, it introduces a potential single point of failure. If the Orchestrator service becomes unavailable, the entire DDI-checking functionality of the application will cease, even if the individual backend services remain operational. To mitigate this vulnerability, the deployment strategy on Google Kubernetes Engine (GKE) must ensure high availability for this specific component. The GKE deployment configuration for the Orchestrator will therefore specify a HorizontalPodAutoscaler (HPA) with a minimum of two, and preferably three, replicas. This configuration ensures that the failure of a single pod will not result in system-wide downtime, transforming the static architectural diagram into a resilient, production-ready deployment plan.

#### 1.2 Google Cloud Platform (GCP) Service Architecture Configuration

The selection of GCP as the target platform was justified in MCR 2 based on its powerful, managed AI and container orchestration services.1 The following provides the specific configuration details for each core service.

* **Google Kubernetes Engine (GKE):** The microservices-based application will be deployed on a GKE Standard cluster. A regional cluster will be used to provide high availability for the control plane across multiple zones, balancing cost and resilience.2 The cluster will be configured with distinct node pools to optimize resource allocation and cost. General-purpose services (Orchestrator, Database Service, etc.) will run on a node pool using cost-effective e2-standard-4 machine types. A separate, dedicated node pool will be configured for the NLP Inference Service, utilizing machine types with attached NVIDIA T4 GPUs to ensure the model has dedicated hardware for low-latency inference without competing for resources with other services.3
* **Vertex AI Endpoint:** To align with the MLOps pipeline designed in MCR 2, the fine-tuned PubMedBERT model will be deployed as a managed Vertex AI Endpoint.1 This approach decouples the model-serving lifecycle from the main application, allowing for independent updates, versioning, and scaling. The endpoint will be configured to use an n1-standard-4 machine type with a single NVIDIA T4 accelerator, providing a balance of performance and cost for real-time prediction requests.5 This configuration provides a scalable, production-ready HTTPS endpoint that the GKE-hosted Orchestrator can call for inference.
* **Cloud SQL for PostgreSQL:** The system's relational database, which stores structured drug information from sources like DrugBank and RxNorm, will be implemented using Cloud SQL for PostgreSQL.1 The "Enterprise" edition will be selected, and it will be deployed in a high-availability (HA) configuration. This creates a standby instance in a different zone for automatic failover, ensuring data durability and minimizing downtime.6 The initial instance will be provisioned with 2 vCPUs, 8 GB of RAM, and 50 GB of SSD storage, a configuration sufficient for the initial data load and query volume, with the ability to scale as needed.
* **Google Cloud Storage (GCS):** GCS serves as the foundational storage layer for the entire MLOps lifecycle.1 A multi-bucket strategy will be employed to logically separate data based on its purpose and access patterns 8:
  + mz02-raw-data: A Standard storage class bucket for housing the raw DDIExtraction 2013 corpus and any future datasets. Standard class is chosen for its high performance, suitable for frequent access during data preprocessing.9
  + mz02-processed-data: A Standard storage class bucket to store the model-ready tensors and other processed data artifacts generated by the data pipeline.
  + mz02-model-artifacts: A Standard storage class bucket for versioned storage of trained model checkpoints and other artifacts produced by Vertex AI Training jobs.
  + mz02-backups: A Coldline storage class bucket for cost-effective, long-term storage of periodic Cloud SQL database backups. Coldline offers the lowest storage cost for data accessed less than once a quarter.9

### 2. Comprehensive Cost Estimation Model

This section provides a detailed financial forecast for both the one-time costs associated with AI model training and the recurring monthly costs of operating the deployed system on GCP. This analysis moves beyond the preliminary research of MCR 1 to a granular, bottom-up cost model that enables strategic financial planning and optimization.1

#### 2.1 Cost Estimation Methodology and Assumptions

The cost model is built from the ground up, calculating the estimated cost for each GCP service based on the finalized architecture. These individual costs are then aggregated to provide a total project cost estimate.

* **Methodology:** All pricing is based on the official, publicly available rates for the us-central1 (Iowa) region to ensure consistency and comparability.2 The model will first calculate a baseline cost using pay-as-you-go (on-demand) pricing and then present optimized scenarios.
* **Assumptions:** The following assumptions underpin the cost model:
  + **Training Phase:** The hyperparameter tuning strategy requires 10 full training and evaluation runs to identify the optimal model configuration.1
  + **Operational Phase:** The production environment is assumed to run continuously, 24 hours a day, 7 days a week, for a total of 730 hours per month.
  + **Usage Volume:** The system is projected to handle a moderate load of 10,000 DDI queries per month.
  + **Data Storage:** The total data footprint across GCS and Cloud SQL is estimated to be 100 GB.
  + **Billing and Credits:** The model will account for the GCP Free Tier, which includes a $300 initial credit for new accounts and ongoing free usage for certain services, such as the management fee for a single GKE cluster.2

#### 2.2 AI Model Training and Experimentation Costs

The most significant one-time cost is the training and hyperparameter tuning of the core NLP model. The choice of GPU hardware is the primary driver of this cost.

* **Hardware Selection Analysis (NVIDIA T4 vs. A100):**
  + **NVIDIA T4:** A power-efficient, lower-cost GPU optimized for inference but also capable of training. Its on-demand price on Vertex AI is approximately $0.40 per hour. While cost-effective on an hourly basis, its lower computational power results in significantly longer training times.5
  + **NVIDIA A100:** A high-performance, data-center-grade GPU designed for intensive AI training workloads. Its on-demand price is substantially higher at approximately $2.93 per hour. However, its Ampere architecture and Tensor Cores can accelerate BERT model training by a factor of 3-5x compared to a T4, drastically reducing the time required for each training run and enabling faster iteration during experimentation.5

The critical decision involves a trade-off between hourly cost and total time-to-result. For a project involving multiple training runs for hyperparameter tuning, the faster completion time of the A100 can lead to a comparable, or even lower, total cost while significantly accelerating the research and development cycle.

* Spot VM Optimization Strategy:  
  A key strategy for mitigating training costs is the use of Spot VMs. These are excess Compute Engine capacity offered at a steep discount of 60-91% compared to on-demand prices.15 The primary trade-off is that these VMs can be preempted (shut down) by Google at any time if the capacity is needed elsewhere. However, Vertex AI Training is designed to handle this; it can automatically restart a preempted job. To leverage this effectively, the training script must implement checkpointing—periodically saving the model's state. This ensures that a restarted job can resume from the last checkpoint rather than starting from scratch, making the training process fault-tolerant and ideally suited for the cost-saving potential of Spot VMs.16

The following table presents a comparative analysis for a single training run, illustrating the cost-performance trade-offs.

| **GPU Type** | **On-Demand Hourly Rate ($)** | **Estimated Training Hours** | **Total On-Demand Cost per Run ($)** | **Spot VM Hourly Rate (Avg. 75% Discount) ($)** | **Total Spot VM Cost per Run ($)** |
| --- | --- | --- | --- | --- | --- |
| **NVIDIA T4** | $0.40 | 12 | $4.80 | $0.10 | $1.20 |
| **NVIDIA A100 40GB** | $2.93 | 3 | $8.79 | $0.73 | $2.19 |

Based on this analysis, while the A100 has a higher hourly rate, its superior performance makes the total cost per run competitive with the T4, especially when using Spot VMs. For the 10 required hyperparameter tuning runs, the total training cost using A100 Spot VMs would be approximately $21.90, plus the associated Vertex AI management fees. This approach provides the best balance of speed and cost-efficiency.

#### 2.3 Production Deployment and Operational Costs

The following table details the estimated recurring monthly costs for running the fully deployed clinical decision support system.

| **GCP Service** | **Configuration Details** | **Estimated Monthly Cost ($)** |
| --- | --- | --- |
| **Google Kubernetes Engine (GKE)** | 1 Regional Cluster, 2 x e2-standard-4 nodes (On-Demand) | $224.00 |
| **Vertex AI Endpoint** | 1 x n1-standard-4 with 1 NVIDIA T4 GPU (On-Demand) | $450.00 |
| **Cloud SQL for PostgreSQL** | 1 x HA Instance (2 vCPU, 8 GB RAM, 50 GB SSD) (On-Demand) | $125.00 |
| **Google Cloud Storage (GCS)** | 100 GB (90% Standard, 10% Coldline) | $2.20 |
| **Networking (Data Egress)** | Estimated 50 GB/month | $6.00 |
| **Total Estimated Monthly Cost** |  | **$807.20** |

This baseline estimate indicates that the primary cost drivers for the operational system are the continuously running compute resources for the Vertex AI Endpoint and the GKE cluster.

#### 2.4 Cost Optimization and Budgeting Strategy

To manage and reduce the ongoing operational costs, several strategies will be implemented.

* **Committed Use Discounts (CUDs):** The GKE worker nodes and the Cloud SQL instance represent stable, predictable workloads that run 24/7. By purchasing a 1-year or 3-year Committed Use Discount for these resources, savings of up to 52% can be realized.3 Applying a 3-year CUD could reduce the combined GKE and Cloud SQL cost from $349.00 to approximately $167.52 per month, lowering the total monthly operational cost significantly.
* **Budgeting and Alerts:** A formal budget will be created within the GCP Billing account. Billing alerts will be configured to automatically notify the project team when spending exceeds predefined thresholds (e.g., 50%, 90%, and 100% of the monthly budget). This proactive monitoring is essential for preventing unexpected cost overruns.
* **Resource Right-Sizing:** After an initial operational period (e.g., one month), the resource utilization metrics for the GKE node pools and the Cloud SQL instance will be analyzed. If the services are consistently underutilized (e.g., average CPU usage is below 30%), the machine types will be downsized to a more appropriate configuration. This practice of right-sizing ensures that the project only pays for the capacity it actually needs.

### 2. Comprehensive Cost Estimation Model (Student Project Focus)

#### This section provides a detailed and realistic financial forecast for a student capstone project, covering both the one-time costs for AI model training and the operational costs for development, testing, and demonstration on GCP. This analysis presents a budget-conscious "Student Path" that prioritizes cost-efficiency while still meeting all project requirements.

#### 2.1 Cost Estimation Methodology and Assumptions

#### The cost model is built from the ground up, calculating the estimated cost for each GCP service based on an architecture optimized for a student project's lifecycle.

#### Methodology: All pricing is based on the official, publicly available on-demand rates for the us-central1 (Iowa) region to ensure consistency.

#### Assumptions: The following revised assumptions underpin this cost model:

#### Training Phase: The hyperparameter tuning strategy still requires 10 full training and evaluation runs to identify the optimal model configuration. This one-time cost is already highly optimized.

#### Operational Phase: The system will not run 24/7. Instead, it will be used on-demand during development sprints (e.g., 8 hours/day, 5 days/week) and for a continuous one-week period for final integration and demonstration. This is the most significant change from a production model.

#### Data Storage: The total data footprint is estimated to be a modest 10-20 GB, sufficient for the DDIExtraction 2013 corpus and model artifacts.

#### Billing and Credits: This model heavily relies on Google Cloud's generous programs for new users. The entire budget is framed by the $300 in free credits available to new accounts, which can be used on any GCP service. It also leverages the GKE free tier, which provides a monthly credit that covers the management fee for one Zonal cluster.

#### 2.2 AI Model Training and Experimentation Costs

#### The one-time cost for training the NLP model remains the same as the production estimate, as this phase was already designed for maximum cost-efficiency using Spot VMs.

#### Hardware Selection and Spot VM Strategy: As established previously, using a high-performance NVIDIA A100 GPU on a Spot VM via Vertex AI Training provides the best balance of speed and cost. Spot VMs offer discounts of 60-91% on compute resources, and Vertex AI's ability to handle preemptions by resuming from checkpoints makes this an ideal strategy for fault-tolerant training jobs.

#### The following table, retained from the initial analysis, confirms the cost-effectiveness of this approach for the 10 required training runs.

| GPU Type | On-Demand Hourly Rate ($) | Estimated Training Hours (per run) | Spot VM Hourly Rate (Avg. 75% Discount) ($) | Total Spot VM Cost per Run ($) |
| --- | --- | --- | --- | --- |
| NVIDIA A100 40GB | $2.93 | 3 | $0.73 | $2.19 |

#### For the 10 required hyperparameter tuning runs, the total one-time training cost is estimated to be approximately $21.90. This highly optimized cost is well within the project's budget and will be the first charge against the $300 free credit.

#### 2.3 Project Development and Demonstration Costs

#### This section details the recurring costs based on a student project lifecycle, fundamentally differing from a 24/7 production model. The primary cost-saving strategy is to use smaller, non-redundant services and to run them only when actively needed.

#### Architectural Changes for Cost Optimization:

#### GKE Cluster: A single Zonal cluster will be used instead of a Regional one, making its management fee eligible to be fully covered by the GKE free tier. The primary worker node will be a small e2-medium instance.

#### AI Model Serving: The dedicated, always-on Vertex AI Endpoint (the largest cost in the production model) is eliminated. Instead, the model will be served from within the GKE cluster on a GPU-enabled node pool. This node pool will be configured to have zero nodes by default and will only be scaled up to one node during active testing or demonstration, converting a fixed monthly cost into a small, on-demand hourly cost.

#### Cloud SQL Database: The High Availability (HA) instance is replaced with a single, non-HA db-f1-micro instance, the smallest and most cost-effective option suitable for development.

#### The following tables model the costs for two realistic scenarios.

#### Scenario A: Typical Development Week (40 hours of active use)

| GCP Service | Configuration Details | On-Demand Hourly Rate ($) | Total Cost for 40 Hours ($) |
| --- | --- | --- | --- |
| GKE Cluster Management | 1 Zonal Cluster | $0.10 | $0.00 (Covered by Free Tier) [4] |
| GKE Worker Node | 1 x e2-medium (2 vCPU, 4 GB RAM) | $0.034 | $1.36 |
| Cloud SQL for PostgreSQL | 1 x db-f1-micro (Shared vCPU, 0.6 GB RAM) | $0.015 | $0.60 |
| Cloud Storage & Networking | ~10 GB | (Negligible) | ~$0.10 |
| Total Estimated Weekly Cost |  |  | ~$2.06 |

#### Scenario B: Final Demonstration Week (168 hours continuous + 8 hours GPU) This assumes the base infrastructure runs for a full week, with the expensive GPU node only active for 8 total hours of final testing and demonstration.

| GCP Service | Configuration Details | On-Demand Hourly Rate ($) | Total Cost for Scenario ($) |
| --- | --- | --- | --- |
| GKE Cluster Management | 1 Zonal Cluster | $0.10 | $0.00 (Covered by Free Tier) [4] |
| GKE Worker Node | 1 x e2-medium (168 hours) | $0.034 | $5.71 |
| Cloud SQL for PostgreSQL | 1 x db-f1-micro (168 hours) | $0.015 | $2.52 |
| GKE GPU Node (Demo Only) | 1 x n1-standard-4 + 1 NVIDIA T4 GPU (8 hours total) | ~$0.57 | $4.56 |
| Cloud Storage & Networking | ~20 GB | (Negligible) | ~$0.50 |
| Total Estimated Demo Week Cost |  |  | ~$13.29 |

#### Export to Sheets

#### 2.4 Final Budget and Cost Management Strategy

#### By adopting a student-centric operational model, the project's total costs are drastically reduced and fall entirely within the scope of Google Cloud's introductory free credits.

#### Total Project Cost Summary

| Cost Component | Estimated Total Cost ($) | Covered by $300 Free Credit? |
| --- | --- | --- |
| One-Time AI Model Training | ~$22.00 | Yes |
| Development & Testing (e.g., 4 weeks) | ~$8.24 | Yes |
| Final Demo Week | ~$13.29 | Yes |
| Grand Total Estimated Cost | ~$43.53 | Yes |
| Actual Out-of-Pocket Expense |  | $0.00 |

#### To ensure costs remain within the free credit allowance, the following management strategies will be implemented:

#### Leverage Free Credits and Tiers: The primary strategy is to utilize the $300 new customer credit to cover all expenses. The use of a Zonal GKE cluster ensures the management fee is also covered by the monthly free tier credit.

#### On-Demand Resource Management: All services, especially the costly GPU node pool in GKE, will be manually shut down or scaled to zero when not in active use. This is the most effective way to control costs in a development context.

#### Budgeting and Alerts: A formal budget will be set in the GCP Billing account for an amount well below the $300 credit (e.g., $100). Billing alerts will be configured to send notifications at 50%, 90%, and 100% of this budget, providing an early warning system to prevent any unexpected consumption of the free credits.

#### Works cited

1. lowlevelarcdesignreport3.drawio (2).pdf
2. Google Kubernetes Engine pricing | Google Cloud, accessed October 30, 2025, <https://cloud.google.com/kubernetes-engine/pricing>
3. Google Kubernetes Engine (GKE) – The Definitive Pricing Guide - Economize Cloud, accessed October 30, 2025, <https://www.economize.cloud/blog/google-kubernetes-engine-pricing-guide/>
4. GCP Kubernetes Pricing - Cost Breakdown & Savings Tips - Pump, accessed October 30, 2025, <https://www.pump.co/blog/gcp-kubernetes-pricing>
5. Vertex AI pricing | Google Cloud, accessed October 30, 2025, <https://cloud.google.com/vertex-ai/pricing>
6. Google Cloud SQL: Pricing and Cost Optimization - ProsperOps, accessed October 30, 2025, <https://www.prosperops.com/blog/google-cloud-sql-pricing-and-cost-optimization/>
7. Cloud SQL pricing | Google Cloud, accessed October 30, 2025, <https://cloud.google.com/sql/pricing>
8. Cloud Storage pricing - Google Cloud, accessed October 30, 2025, <https://cloud.google.com/storage/pricing>
9. Pricing examples | Google Cloud Documentation, accessed October 30, 2025, <https://cloud.google.com/storage/pricing-examples>
10. Free cloud features and trial offer | Google Cloud Free Program, accessed October 30, 2025, <https://docs.cloud.google.com/free/docs/free-cloud-features>
11. What are the key differences between NVIDIA A100 and T4 GPUs in ..., accessed October 30, 2025, [https://massedcompute.com/faq-answers/?question=What%20are%20the%20key%20differences%20between%20NVIDIA%20A100%20and%20T4%20GPUs%20in%20terms%20of%20performance%20and%20power%20efficiency?](https://massedcompute.com/faq-answers/?question=What+are+the+key+differences+between+NVIDIA+A100+and+T4+GPUs+in+terms+of+performance+and+power+efficiency?)
12. NVIDIA T4 vs. NVIDIA A100 Comparison: Which GPU Should You Choose for AI and Data Center Workloads? - server-parts.eu, accessed October 30, 2025, <https://www.server-parts.eu/post/nvidia-t4-vs-a100-gpu-comparison-ai-deep-learning-data-centers>
13. How does the training time of BERT models compare on T4 and A100 GPUs?, accessed October 30, 2025, [https://massedcompute.com/faq-answers/?question=How%20does%20the%20training%20time%20of%20BERT%20models%20compare%20on%20T4%20and%20A100%20GPUs?](https://massedcompute.com/faq-answers/?question=How+does+the+training+time+of+BERT+models+compare+on+T4+and+A100+GPUs?)
14. DEEP LEARNING PERFORMANCE GUIDE - M Computers, accessed October 30, 2025, <https://mcomputers.cz/wp-content/uploads/2020/12/3_NVIDIA_Ralph_Hinsche.pdf>
15. Pricing | Spot VMs | Google Cloud, accessed October 30, 2025, <https://cloud.google.com/spot-vms/pricing>
16. Use Spot VMs with training | Vertex AI - Google Cloud Documentation, accessed October 30, 2025, <https://docs.cloud.google.com/vertex-ai/docs/training/use-spot-vms>
17. Articles | DrugBank Help Center, accessed October 30, 2025, <https://dev.drugbank.com/guides/drug_interactions/parameters>
18. Drug Interaction Checker | DrugBank Online, accessed October 30, 2025, <https://go.drugbank.com/drug-interaction-checker>
19. Temperature scaling - AWS Prescriptive Guidance, accessed October 30, 2025, <https://docs.aws.amazon.com/prescriptive-guidance/latest/ml-quantifying-uncertainty/temp-scaling.html>
20. Calibrating Neural Networks - Geoff Pleiss, accessed October 30, 2025, <https://geoffpleiss.com/blog/nn_calibration.html>
21. gpleiss/temperature\_scaling: A simple way to calibrate your neural network. - GitHub, accessed October 30, 2025, <https://github.com/gpleiss/temperature_scaling>
22. EFS-OpenSource/calibration-framework: The net:cal ... - GitHub, accessed October 30, 2025, <https://github.com/EFS-OpenSource/calibration-framework>
23. saurabhgarg1996/calibration: Simple implementation of Temperature and Vector Scaling in python - GitHub, accessed October 30, 2025, <https://github.com/saurabhgarg1996/calibration>
24. Classification: Accuracy, recall, precision, and related metrics ..., accessed October 30, 2025, <https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall>
25. How to Handle Imbalanced Datasets in Predictive Modeling for Accurate Results?, accessed October 30, 2025, <https://learn.microsoft.com/en-us/answers/questions/2243657/how-to-handle-imbalanced-datasets-in-predictive-mo>
26. Imbalanced class distribution and performance evaluation metrics: A ..., accessed October 30, 2025, <https://pmc.ncbi.nlm.nih.gov/articles/PMC10688675/>
27. ROC AUC vs Precision-Recall for Imbalanced Data - MachineLearningMastery.com, accessed October 30, 2025, <https://machinelearningmastery.com/roc-auc-vs-precision-recall-for-imbalanced-data/>
28. ROC and precision-recall with imbalanced datasets, accessed October 30, 2025, <https://classeval.wordpress.com/simulation-analysis/roc-and-precision-recall-with-imbalanced-datasets/>
29. codesignal.com, accessed October 30, 2025, <https://codesignal.com/learn/courses/building-and-evaluating-text-classifiers-in-python/lessons/mastering-cross-validation-for-text-classification-in-python#:~:text=The%20most%20common%20type%20of,subset%20as%20the%20validation%20set.&text=In%20Python%2C%20we%20can%20easily,specifically%20the%20cross_val_score()%20function.>
30. Understand K-Fold Cross-Validation:A Step-by-Step Guide ..., accessed October 30, 2025, <https://www.digitalocean.com/community/tutorials/k-fold-cross-validation-python>
31. 3.1. Cross-validation: evaluating estimator performance - Scikit-learn, accessed October 30, 2025, <https://scikit-learn.org/stable/modules/cross_validation.html>
32. A taxonomy for advancing systematic error analysis in multi-site ..., accessed October 30, 2025, <https://pmc.ncbi.nlm.nih.gov/articles/PMC11187420/>
33. New Benchmarks Envision the Future of AI in Healthcare | Scale, accessed October 30, 2025, <https://scale.com/blog/healthcare-benchmarks>
34. Evaluating AI in context: Which LLM is best for real health care needs? - Stanford Medicine, accessed October 30, 2025, <https://med.stanford.edu/news/insights/2025/04/ai-artificial-intelligence-evaluation-algorithm.html>
35. How AI-based simulation testing is revolutionising healthcare - Pearson VUE, accessed October 30, 2025, <https://www.pearsonvue.com/us/en/about/news/highlights/how-ai-based-simulation-testing-is-revolutionising-healthcare.html>
36. The Integration of Artificial Intelligence into Clinical Practice - MDPI, accessed October 30, 2025, <https://www.mdpi.com/2813-0464/3/1/2>
37. Monitoring performance of clinical artificial intelligence in health care: a scoping review, accessed October 30, 2025, <https://pmc.ncbi.nlm.nih.gov/articles/PMC11630661/>