Advanced AI Diagnostics: Vertebral Classification Using AWS

Module Assignment for

CS5024 - Theory and Practice of Advanced AI Ecosystems

Student Name: **ANANYA KRITHIKA THYAGARAJAN**

Student ID: **23187123**

Revision Timestamp: 02/05/2024:20:15

Contents

Abstract

This project aims to address the critical demand for accurate and effective spine diagnostics by creating a machine learning solution that categorises vertebral column data into normal and problematic groups. The project aims to utilise artificial intelligence to enhance the accuracy of spine diagnostics and streamline operational efficiency, in response to the constraints faced by present diagnostic techniques. The solution incorporates a range of AWS services, including SageMaker for training and deploying machine learning models, S3 for efficient data storage and administration, Lambda for automating operational activities, and CloudWatch for continuous performance monitoring. The architecture is specifically built to have the ability to grow and adapt without losing efficiency or strength, resulting in a cutting-edge diagnostic tool that improves the procedures of making clinical decisions in the medical profession. The implementation seeks to provide a dependable and automated system that assists healthcare providers in delivering quicker and more precise spinal diagnoses.

Introduction

The vertebral column, or spine, is the basic anatomical and neurological support structure of humans. The central nervous system, which regulates body functions, is housed in the skeletal system, which also serves to support the body. These crucial duties might be hampered by age-related spinal issues and birth anomalies, leading to physical limitations and persistent pain. The way in which these illnesses affect one's quality of life highlights the necessity of efficient diagnostic treatments.

Radiographs have long been used by medical professionals to diagnose spinal diseases. This method, while simple, has a lot of disadvantages. Data analysis in radiography is subjective and contingent on the expertise and experience of the observer.

Additionally, using this procedure could result in an incorrect diagnosis, delaying treatment.

Diagnostics in medicine is one of the many domains where machine learning is significant and potent. Because of its great accuracy in classification, it helps enhance diagnoses. Evaluations produced by machine learning algorithms are more reliable and accurate because they can analyse intricate patterns in visual data that are invisible to humans. Artificial Intelligence is a potent tool for increasing diagnostic precision and decreasing errors.

This project will use Amazon Web Services (AWS) to build an AI-based diagnostic model, leveraging technical advancements. CloudWatch for monitoring, Lambda for operational automation, S3 for data storage, and SageMaker for machine learning are all easily integrated with AWS. These interactions improve accessibility and scalability of the model and offer a strong foundation for real-time diagnostics. This programme enhances the subjectivity, availability, and accuracy of spinal diagnoses through the use of AI and cloud computing.

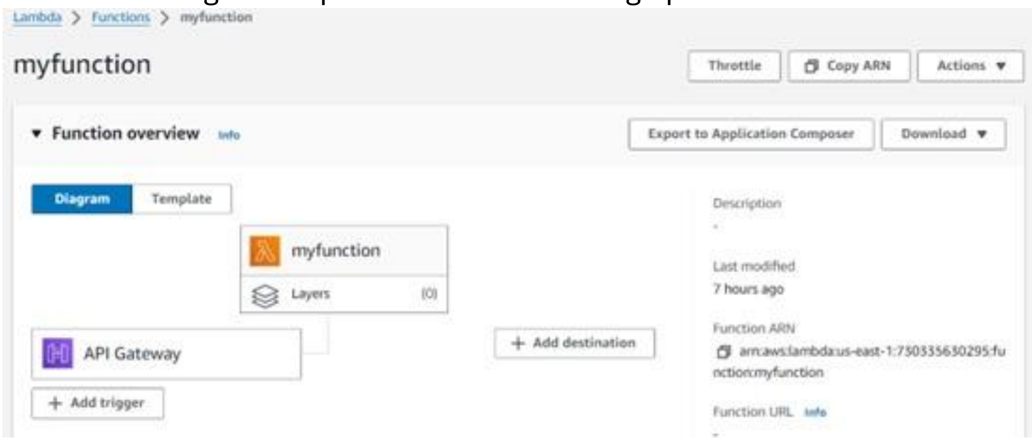
AI Ecosystem Architecture Used

The architectural diagram illustrates a complex AWS cloud configuration that effectively manages and processes client requests through a sequence of AWS services, principally within a VPC (Virtual Private Cloud) located in the North Virginia region.

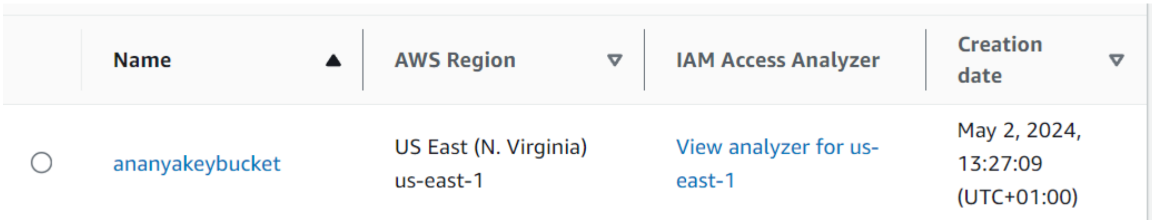
The process commences when a client transmits an HTTP POST request, which is subsequently received by the Amazon API Gateway. The API Gateway acts as the primary entry point for all incoming API calls, efficiently overseeing and directing these requests. After the API Gateway handles the request, it subsequently transfers it to AWS Lambda.

AWS Lambda, a compute service that operates without the need for servers, then runs the code according to the parameters of the request. This phase is essential because it enables the processing of data without the need to provision or manage servers.

While being executed, Lambda engages with several AWS services. An important interaction is with Amazon SageMaker, which is a comprehensive tool that allows developers to construct, train, and implement machine learning models. Lambda has the capability to initiate activities within SageMaker, such as fetching model predictions or receiving updates.



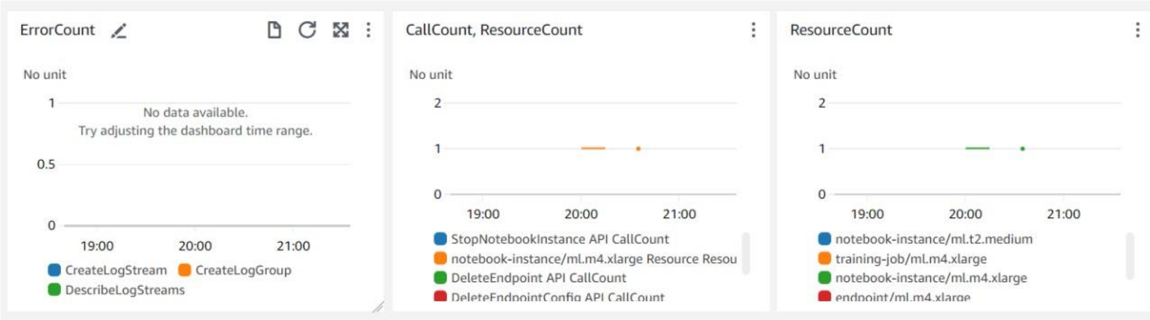
Lambda has the capability to interact with Amazon S3 (Simple Storage Service) as part of its operational responsibilities.
Amazon S3 is used to store and manage datasets in the form of zip files. The Lambda function has the capability to either retrieve data from S3 or save the output back into it, depending on the logic of the application.



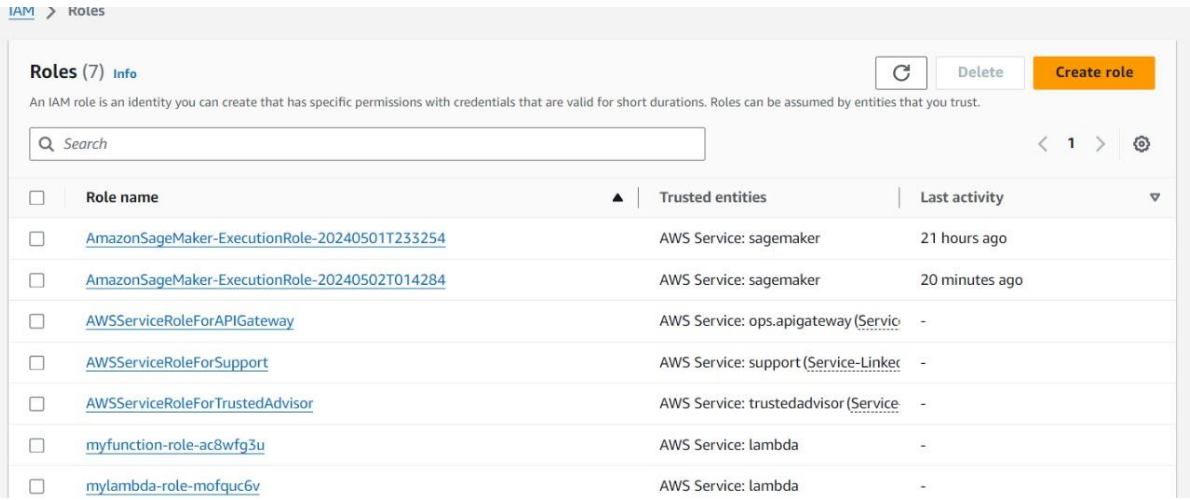
Name	AWS Region	IAM Access Analyzer	Creation date
ananyakeybucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	May 2, 2024, 13:27:09 (UTC+01:00)

Amazon CloudWatch is responsible for both monitoring and logging, capturing and tracking the operational metrics of the application at the same time. This allows for monitoring performance and conducting operational health assessments.

CloudWatch has the capability to activate alarms using AWS SNS (Simple Notification Service) when certain thresholds are reached. This notifies administrators about potential problems or the requirement for intervention.



Furthermore, AWS IAM (Identity and Access Management) plays a vital role in this architecture by overseeing permissions and guaranteeing that each service and user possess the appropriate access levels for resources.

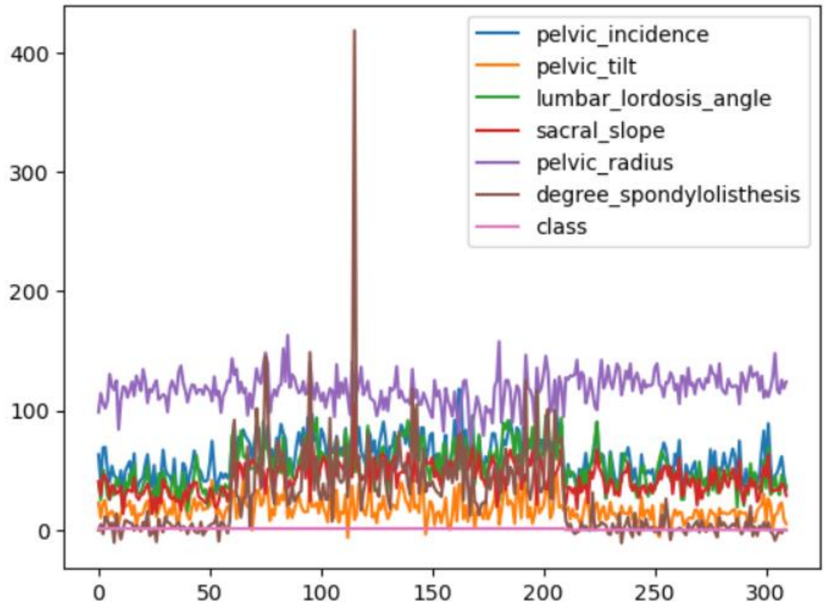


The entire procedure may also include AWS Auto Scaling, which dynamically adjusts the allocation of resources based on demand or predefined parameters, ensuring efficient management of fluctuating workload levels.

This architecture effectively manages data processing and model management in a scalable, serverless environment. It also offers reliable operational monitoring and security management.

Model Description

The Jupyter notebook model is specifically created to forecast spinal problems by utilising vertebral column data that has been processed through AWS infrastructure. This model is constructed via the XGBoost method, assisted by Amazon SageMaker, which offers a platform for training and deploying machine learning models.



The data, which is housed in Amazon S3, contains biomechanical parameters such as pelvic incidence, pelvic tilt, lumbar lordosis angle, sacral slope, pelvic radius, and degree of spondylolisthesis. The dataset is initially subjected to preprocessing, which involves dividing it into separate sets for training, validation, and testing. This is done to enhance the dataset's resilience and avoid overfitting. Features are derived from the ARFF file format, which is a widely used format for storing datasets that have a large number of empty or zero values.

The AWS API is utilised to retrieve the data, which is subsequently pre-processed within a Jupyter notebook environment. Finally, the treated data is posted back to S3 in CSV format. XGBoost in SageMaker can directly access the training and validation datasets from S3, ensuring efficient data management and scalability.

The process of model training requires the specification of different hyperparameters, such as the number of iterations and the objective measure (AUC for binary classification). After completing the training process, the model is employed to generate predictions on the test dataset, evaluating the existence or nonexistence of vertebral anomalies.

```
train_channel = sagemaker.inputs.TrainingInput(
    "s3://{}/{}/train/".format(bucket,prefix,train_file),
    content_type='text/csv')

validate_channel = sagemaker.inputs.TrainingInput(
    "s3://{}/{}/validate/".format(bucket,prefix,validate_file),
    content_type='text/csv')

data_channels = {'train': train_channel, 'validation': validate_channel}
```

```
xgb_model.fit(inputs=data_channels, logs=False)
```

```
INFO:sagemaker:Creating training-job with name: sagemaker-xgboost-2024-05-02-14-57-19-776

2024-05-02 14:57:19 Starting - Starting the training job.........
2024-05-02 14:58:11 Starting - Preparing the instances for training..........
2024-05-02 14:59:14 Downloading - Downloading input data......
2024-05-02 14:59:49 Downloading - Downloading the training image......
2024-05-02 15:00:30 Training - Training image download completed. Training in progress....
2024-05-02 15:00:50 Uploading - Uploading generated training model.
2024-05-02 15:01:01 Completed - Training job completed
```

The results, namely the probability of abnormalities, undergo additional processing to categorise them as either normal or abnormal, depending on a selected threshold.

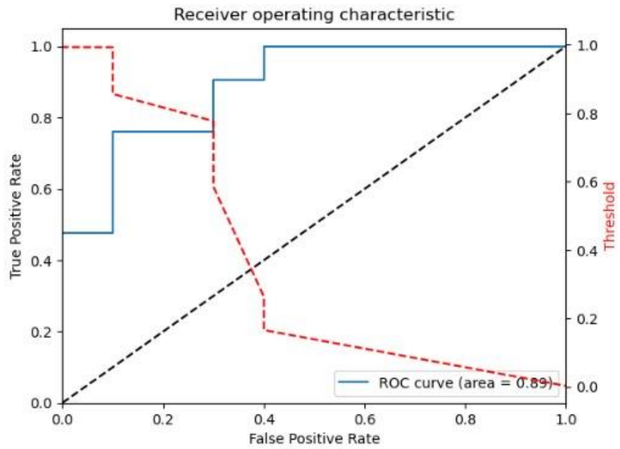
The complete configuration highlights the use of many AWS services to optimise the workflow from data storage, model training, to prediction generation, demonstrating a scalable and resilient solution for medical diagnostic applications. This automated and fast method is advantageous for managing intricate datasets and making predictions on a wide scale in the healthcare field.

Model Parameters

The value of "num_rounds" is set to 42, which represents the number of boosting rounds or trees to be constructed.

The evaluation metric used is 'auc', with a focus on assessing the model's performance based on the area under the ROC curve.

The objective is defined as 'binary:logistic', which specifies the learning problem and the accompanying learning objective.



Training and Validation:
The data is divided into a 70% training set and a 30% validation set to ensure effective learning while avoiding overfitting. The hyperparameters of the model were optimised using AWS SageMaker's automatic model tuning, resulting in improved model accuracy.

Accuracy, Precision and Recall:

Accuracy is a measure of the proportion of correctly identified classifications, including both positive and negative ones. It is a simple and precise metric that quantifies the accuracy of the model in all its predictions.

Recall, also known as sensitivity, pertains to the model's capacity to accurately detect true positives from the given data. This statistic is essential in situations where the consequences of failing to identify a positive case are much more severe than incorrectly classifying a negative case, as is the case with disease screening.

Precision measures the proportion of correctly identified positive cases out of all the cases that were projected to be positive. It is crucial, especially in situations when the negative outcomes of inaccurate identifications are significant, such as in spam detection where genuine emails may be mistakenly identified as spam.

The F1-score is a statistic that combines precision and recall by calculating their harmonic mean. It offers a comprehensive measure that strikes a compromise between precision and recall. This measure is particularly beneficial when you need to include both metrics and there is an unequal distribution of classes that could affect the accuracy metre.

Below is a concise description of each metric presented in the output:

Precision (ACC): The value is 83.879674139549. The percentage provided is the total proportion of accurate results, including both true positives and true negatives, out of the total number of instances analysed. It is a comprehensive indicator of the model's overall performance across all categories.

The sensitivity or true positive rate (TPR) is 90.47619047619048%. This figure represents the model's accuracy in correctly identifying positive events. A high sensitivity indicates that the model have a strong ability to detect true positives, hence minimising the occurrence of false negatives.

The specificity or true negative rate (TNR) is 70.0%. This value represents the proportion of actual negative cases that are correctly identified as negative. Specificity quantifies the model's capacity to accurately detect and classify negative instances. Within this particular framework, it is evident that the model accurately detects 70% of the negative instances, hence preventing the occurrence of false positives.

Precision: Value: 86.36% Explanation: Precision is a measure of the accuracy of positive forecasts. It quantifies the accuracy of affirmative identifications. Enhanced precision minimises the likelihood of false positives.

The Negative Predictive Value (NPV) is defined as the percentage of negative results that are actually negative. In this case, the NPV is 77.77777777777779%. It is essential in situations where it is imperative to reduce the likelihood of incorrect negative results.

False Positive Rate (FPR): The value is 30.0%.

Definition: This statistic represents the likelihood that a correct negative prediction will be mistakenly identified as a positive prediction. A lower false positive rate (FPR) indicates a higher level of accuracy in correctly identifying negative cases by the model.

The False Negative Rate (FNR) is defined as 9.523809523809524%. It is the likelihood that a true positive is mistakenly identified as a negative. Reducing missed affirmative cases is more desirable when values are lower.

The False Discovery Rate (FDR) is 13.636363636363635%. It is the percentage of false positives in relation to the total number of positive predictions. A lower false discovery rate (FDR) indicates that a model has a high capacity to accurately detect positive events.

```python
# Sensitivity, hit rate, recall, or true positive rate
Sensitivity  = float(TP)/(TP+FN)*100
print(f"Sensitivity or TPR: {Sensitivity}%")
print(f"There is a {Sensitivity}% chance of detecting patients with an abnormality have an abnormality")
```

```
Sensitivity or TPR: 90.47619047619048%
There is a 90.47619047619048% chance of detecting patients with an abnormality have an abnormality
```

```python
# Specificity or true negative rate
Specificity  = float(TN)/(TN+FP)*100
print(f"Specificity or TNR: {Specificity}%")
print(f"There is a {Specificity}% chance of detecting normal patients are normal.")
```

```
Specificity or TNR: 70.0%
There is a 70.0% chance of detecting normal patients are normal.
```

```python
# Precision or positive predictive value
Precision = float(TP)/(TP+FP)*100
print(f"Precision: {Precision}%")
print(f"You have an abnormality, and the probablity that is correct is {Precision}%")
```

```
Precision: 86.36363636363636%
You have an abnormality, and the probablity that is correct is 86.36363636363636%
```

```python
# Negative predictive value
NPV = float(TN)/(TN+FN)*100
print(f"Negative Predictive Value: {NPV}%")
print(f"You don't have an abnormality, but there is a {NPV}% chance that is incorrect" )
```

```
Negative Predictive Value: 77.77777777777779%
You don't have an abnormality, but there is a 77.77777777777779% chance that is incorrect
```

```python
# Fall out or false positive rate
FPR = float(FP)/(FP+TN)*100
print( f"False Positive Rate: {FPR}%")
print( f"There is a {FPR}% chance that this positive result is incorrect.")
```

```
False Positive Rate: 30.0%
There is a 30.0% chance that this positive result is incorrect.
```

```python
# False negative rate
FNR = float(FN)/(TP+FN)*100
print(f"False Negative Rate: {FNR}%")
print(f"There is a {FNR}% chance that this negative result is incorrect.")
```

```
False Negative Rate: 9.523809523809524%
There is a 9.523809523809524% chance that this negative result is incorrect.
```

```python
# False discovery rate
FDR = float(FP)/(TP+FP)*100
print(f"False Discovery Rate: {FDR}%" )
print(f"You have an abnormality, but there is a {FDR}% chance this is incorrect.")
```

```
False Discovery Rate: 13.636363636363635%
You have an abnormality, but there is a 13.636363636363635% chance this is incorrect.
```

The image shows a Python script with code and notes elucidating diverse statistical measures obtained from a confusion matrix of a classification model. These measurements are crucial for comprehending the performance of a model in relation to accuracy, error rates, and precision. Below is a concise description of each metric that is computed:

Sensitivity, also known as Recall or True Positive Rate, quantifies the model's capacity to accurately detect true positives in the dataset. It indicates how effectively the model recognises actual instances of the tested condition.

Specificity, also known as the True Negative Rate, measures the model's accuracy in properly identifying instances where the condition is absent.

Precision: This indicator measures the level of accuracy in the model's positive predictions. Precision is the proportion of true positives to all positive predictions made by the model, which indicates the probability that a positive prediction is accurate.

The Negative Predictive Value (NPV) is a measure of the accuracy of a negative result from the model. It is produced by dividing the number of true negatives by the total number of forecast negatives.

The False Omission Rate (FOR) is the likelihood of a negative result being inaccurate. It is determined by dividing the number of false negatives by the total number of negative outcomes predicted by the model.

The False Positive Rate, also referred to as the probability of a false alarm, represents the frequency at which non-conditions are inaccurately classified as conditions.

The False Negative Rate refers to the likelihood of the model mistakenly classifying a patient with a condition as not having the condition.

The False Discovery Rate (FDR) quantifies the ratio of incorrect positive predictions to the overall number of positive outcomes estimated by the model.

Results and Application

The accuracy of the predictions is 83.87%, which indicates a good level of correctness.

The precision of the model is 86.36%, indicating a high likelihood of successfully classifying vertebral anomalies.

The recall rate of the model is 90.47%, indicating that it successfully identifies the majority of real instances of anomalies.

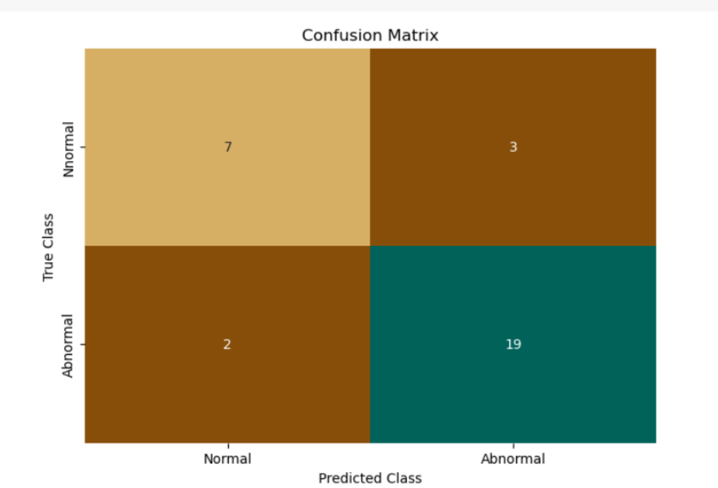
Based on these findings, we can deduce the following:

The model demonstrates more efficacy in detecting "Abnormal" cases compared to "Normal" cases, as indicated by a higher count of true positives.

The incidence of false negatives is relatively minimal, indicating that the model is cautious in forecasting regular outcomes, with a focus on minimising the possibility of overlooking an abnormal occurrence.

The model's accuracy can be determined using the formula (TP+TN)/(TP+TN+FP+FN), which calculates the proportion of correctly classified cases out of all cases.

This matrix offers valuable insights into the model's capabilities and identifies areas that require additional improvement, specifically in lowering false positives and enhancing the identification of normal situations.



Visualizations:

The SageMaker notebook was used to construct a Confusion Matrix and ROC curves in order to visually represent the performance nuances and threshold effects.

Operational Efficiency:

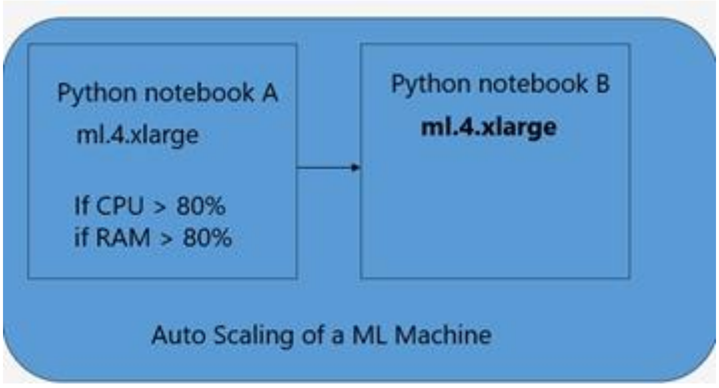
The deployment on AWS showcased cost-effectiveness by efficiently utilising resources only when model predictions were needed, hence minimising idle periods and expenses associated with conventional round-the-clock server operations.

Scalability Considerations

Considering the AWS architecture shown, it is important to consider various factors to ensure that the solution can efficiently scale and effectively handle increasing loads or demands. Here are a few tactics to contemplate:

Disaster Recovery and High Availability: When expanding, make sure that the architecture is able to withstand failures. Deploy important components in many Availability Zones (AZs) to ensure high availability. Additionally, if your application has users worldwide, consider deploying it in multiple regions.

Autoscaling: Make sure that any EC2 instances or other resources that may be scaled in the design utilise AWS Auto Scaling to dynamically alter their capacity, ensuring consistent and predictable performance while minimising expenses.



Lambda Performance: To optimise the execution speed of heavier workloads in AWS Lambda, it is advisable to increase the memory allocation. It is essential to examine the concurrency settings to guarantee that Lambda can automatically adjust its scaling based on the volume of requests.

SageMaker Optimization: Ensure that the model in Amazon SageMaker is optimised for optimal performance. This entails choosing the appropriate instance type for both training and deploying models, as well as utilising SageMaker tools such as Automatic Model Tuning to optimise model parameters. When dealing with large-scale implementations, it is advisable to utilise SageMaker multi-model endpoints to efficiently serve numerous models from a single endpoint.

S3 and Data Management: To effectively manage data growth, it is important to organise your S3 buckets and utilise lifecycle policies to properly archive and manage data. Please consider activating S3 Transfer Acceleration to enhance upload and download speeds between geographically distant locations.
By taking into account these factors, you can guarantee that your AWS infrastructure not only expands efficiently but also maintains a strong, secure, and cost-effective state.

Services and their Purpose
The AWS cloud architecture devised for this project incorporates multiple services that collaborate to enable a smooth transition from data storage to model deployment:

Amazon API Gateway: Acts as the initial access point for clients, facilitating secure HTTP connectivity to backend services. The function directs requests to the suitable AWS Lambda function for processing, guaranteeing secure and efficient access and utilisation of the model for predictions.

AWS Lambda: Automates the process of preparing data, calling the model, and handling the results of the model's predictions. API Gateway triggers lambda functions, which manipulate data from Amazon S3 and perform model predictions using SageMaker.

Amazon SageMaker: SageMaker is a key component of the project since it serves as a platform for hosting the Jupyter notebook used for data exploration and model training. It also handles the management of several machine learning models and automates the process of deploying these models into production.

Amazon S3: Serves as the centralised repository for all project data. S3 buckets are utilised for the storage and retrieval of raw datasets, as well as for storing intermediate outputs and the final outcomes of studies.

IAM (Identity and Access Management): Responsible for the process of verifying and granting permission, guaranteeing secure and designated entry to AWS resources utilised in the project.

Amazon CloudWatch and SNS (Simple Notification Service): Monitor the system's well-being and efficiency, issuing alerts and messages when particular triggers or thresholds are met.

Diagram Explanation

This diagram depicts the interconnected AWS services that enable the smooth functioning of the project. The labelling of each component clearly indicates its function and how it interacts with other services, emphasising the cloud-based workflow that has been specifically built for optimal efficiency and scalability.

The architecture facilitates scalability by leveraging AWS's elastic services. When the amount of data or the frequency of requests rises, services such as Amazon S3 and SageMaker can automatically adapt and scale up with the help of AWS Auto Scaling. This system's scalability guarantees its capacity to accommodate possible extensions, such as the inclusion of more comprehensive patient data or the ability to scale up to a multi-tenant system, where multiple healthcare providers can use the service simultaneously.

Challenges and Solutions

The issue of imbalanced data was resolved by utilising synthetic data generation techniques throughout the training process. The optimisation of AWS resources was achieved by the use of cost-effective resource selection and the practice of shutting down instances when they are not in use.

Future Work and Improvements

Future improvements involve incorporating deep learning models to analyse more intricate patterns in imaging data and developing the service into a comprehensive diagnostic platform that seamlessly connects with electronic health records (EHRs).

Conclusion

This project demonstrates the capacity of AWS and AI to transform medical diagnosis. The successful deployment of a model for classifying vertebral columns showcases how cloud architectures can enable sophisticated AI solutions in the healthcare sector, providing scalability, efficiency, and improved diagnostic precision.

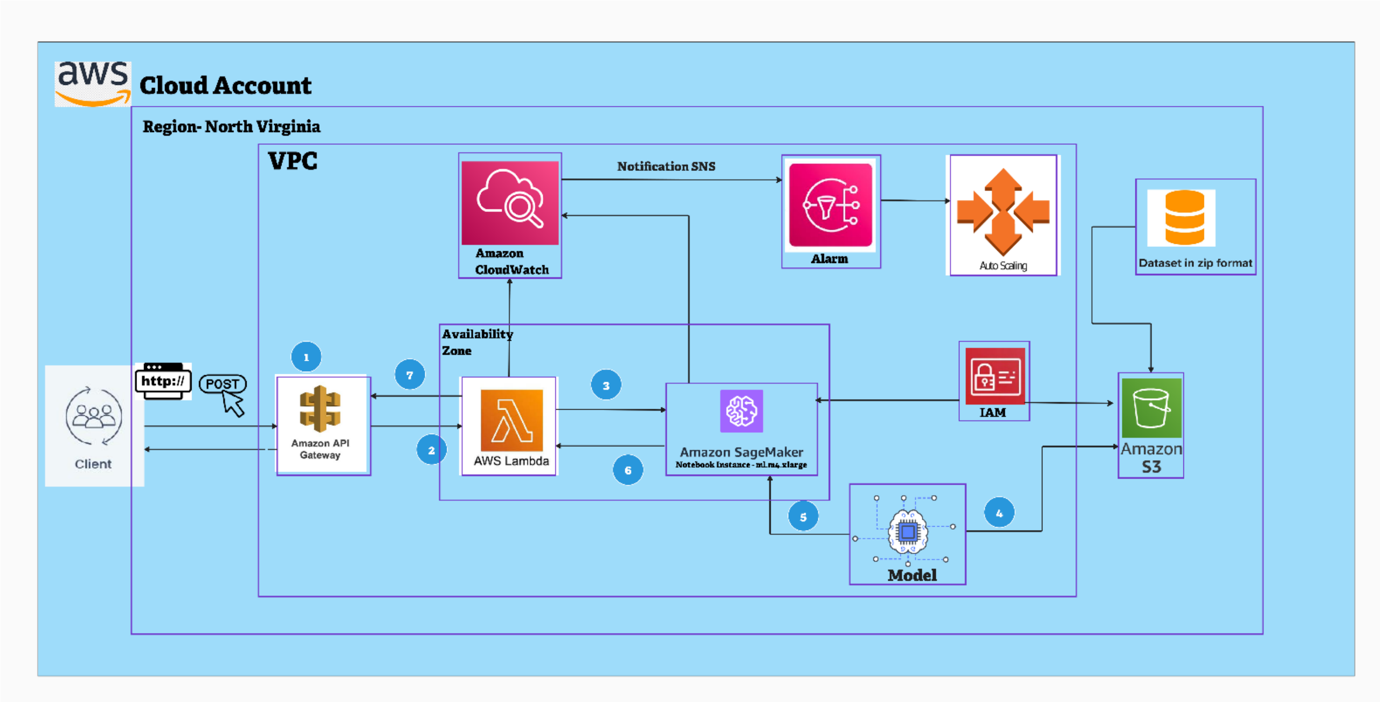


Figure 1: Architecture used to create AI/ML ecosystem for Model X

Potential Expansion of the Ecosystem

a) Improved Data Management:

Adopt Data Lakes: As the amount of data increases, shifting from using S3 as basic storage to deploying AWS Lake Formation can enhance data management by enabling more efficient data cleaning, classification, and querying capabilities.

Integrate Amazon Redshift to enhance analytical capabilities and enable faster and more complicated querying of huge datasets commonly found in expanding machine learning projects.

b) Scalability Enhancements:

Auto Scaling for Lambda: Implement auto-scaling strategies for AWS Lambda to effectively manage surges in request volumes, thereby sustaining optimal performance without the need for manual intervention.

Elastic Load Balancing (ELB) can be implemented in front of the API Gateway to effectively divide incoming traffic over numerous instances or containers. This helps to alleviate the burden on individual resources by evenly distributing the load.

c) Improved Machine Learning Pipeline:

Utilize SageMaker Pipelines: To enhance the management of the machine learning lifecycle, make use of AWS SageMaker Pipelines. This will automate and coordinate your ML workflows, ensuring that they are consistent, reproducible, and scalable.

d) Security Enhancements:

Enhanced IAM Roles: Examine and improve IAM roles and policies to adhere to the principle of least privilege, particularly crucial when the system expands and additional endpoints or services are incorporated.

Implement network segmentation by dividing the Virtual Private Cloud (VPC) into smaller subnets. Use private subnets for backend functions such as SageMaker, Lambda, and data storage, while placing the API Gateway in a public subnet.

e) Reliability and Monitoring:

Implement advanced monitoring metrics and logging using AWS CloudWatch to enhance reliability. - Set up alerts to detect abnormalities or performance concerns. It is advisable to incorporate AWS X-Ray for the purpose of tracing and conducting more comprehensive analysis of applications.

Implement a system of SNS (Simple Notification Service) to improve the use of real-time alerts. This will allow for swift response to any problems with model performance or operational issues.

f) Cost Management:

Deploy AWS Cost Explorer: Efficiently oversee and control expenses while expanding your operations. Use AWS Budgets to establish cost thresholds and notifications to prevent unforeseen spending.

g) Disaster Recovery (DR) involves deploying important components of the application across different AWS regions to ensure both high availability and the ability to recover from a disaster. This guarantees that in the event of a failure in a certain location, the system will continue to operate.

References

Patrick Denny. (2024, February 26). *AWS Cloud Security - Tuesday Week 4 Material*. Retrieved from CS5024 - Theory and Practice of Advanced AI Ecosystems: https://learn.ul.ie/d2l/le/lessons/17937/topics/634587
AWS Official Documentation.
XGBoost Documentation.

Latest research papers on AI in healthcare from PubMed and other medical journals.

Appendix

Additional code snippets, configuration details, and resource links can be found here, providing deeper insights into the project's technical foundations.