

# **Configuration Manual**

MSc Research Project Programme Name

Forename Rajaram Jagadeeswaran Student ID: x22239243

> School of Computing National College of Ireland

Supervisor: Jitendra Kumar Sharma

#### **National College of Ireland**



#### **MSc Project Submission Sheet**

#### **School of Computing**

Student Name:	Rajaram Jagadeeswaran				
Student ID:	X22239243				
Programme:	MSc in Cloud Computing Year:2024				
Module:	Research Project				
Lecturer: Submission Due Date:	16/09/2024				
Due Dute.					
Project Title:	Optimizing Healthcare Framework using Cognitive Computing Techniques in Cloud: A Study on Enhancing Diagnostic Accuracy and Decision-Making				
Word Count:	400 Page Count:10				

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Rajaram Jagadeeswaran		
_			

#### Date: ...16/09/2024.....

#### PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple	
copies)	
Attach a Moodle submission receipt of the online project	
submission, to each project (including multiple copies).	
You must ensure that you retain a HARD COPY of the project,	
both for your own reference and in case a project is lost or mislaid. It is	
not sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office	Use	Only

Unice use Uniy	
Signature:	
Date:	
Penalty Applied (if applicable):	

## **Configuration Manual**

Rajaram Jagadeeswaran Student ID: x22239243

## 1 Prerequisite of AWS Account Setup

First make sure you have an active AWS account (private/ college). If you don't have one, sign up at <u>AWS</u>.

**IAM Roles & Permissions:** Set up required IAM Roles with actual permissions what is needed, which includes:

For adding new policy, need to use <u>AWS Generate Policy</u>. By adding valid ARN for any service and principal, you can be able to create new policies, follow this step throughout in all resources otherwise you can use the policy edit option to manually edit the policy for your need.

AWS Policy Generator					
The AWS Policy Generator is a tool that enables you to create policies that control access to Amazon Web Services (AWS) products and resources. For more informatio policies, see key concepts in Using AWS Identity and Access Management. Here are sample policies.					
Step 1: Select Policy Type					
A Policy is a container for permissions. Queue Policy.	The different types of policies you can c	reate are an IAM Policy, an S3 Bucket Policy, an SNS Topic Pol	icy, a VPC Endpoint Policy,		
Select Type of Policy	S3 Bucket Policy				
Step 2: Add Statement(s)					
A statement is the formal description o	f a single permission. See a description	of elements that you can use in statements.			
Effect	● Allow ○ Deny				
Principal	*				
	Use a comma to separate multiple values.				
AWS Service	Amazon S3				
All Services ('*')	Use multiple statements to add permissions for	more than one service.			
Actions	2 Action(s) Selected				
Amazon Recource Name (ARN)	[				
Anazon Resource Name (ARN)	ARN should follow the following format: arn:aws	::s3:::\${BucketName}/\${KeyName}.			
	Use a comma to separate multiple values.				
	Add Conditions (Optional)				
	Add Statement				
You added the following statements. Click the button below to Generate a policy.					
Principal(s) Effect	Action	Resource	Conditions		
• * Allow	<ul> <li>s3:CreateBucket</li> </ul>	arn:aws:s3:::x22239243-spm/*	None		
Sten 3: Generate Policy					
A policy is a document (written in the	Access Policy Language) that acts as a co	ontainer for one or more statements.			
reporter to a account (mittain in the r	interest inter congreger that allo as a co				
	Generate Policy Start Over				
Figure 1. Generate	Figure 1. Generate policy				

Once policy is generated you can copy it to add in services specifically.

#### 1. SageMaker Execution Role:

This role should have permission to interact with S3 bucket wherever going to be used by SageMaker, Bedrock. If you are using multiple buckets, make sure to create execution roles for each S3 bucket.

• Below figure is the model of existing S3 bucket permission added via specific S3 bucket from permission tab.



Figure 2. S3 Bucket Policy for utilizing in SageMaker

<u>iam</u> >	Roles >	AmazonSageMaker-ExecutionRole-20240728T224074			
Ama SageMak	IZONS	ageMaker-ExecutionRole-20240	728T224074 Info		
Sum	mary				
Creati July 2 Last a 2 c	ion date 28, 2024, 22 activity days ago	1:41 (UTC+01:00)			ARN  ARN  ARN  ARN  Arn.aws.lam::533267211033.role/service-role/AmazonSageMaker-Exe Maximum session duration 1 hour
Perm	issions	Trust relationships Tags Access Advisor Re	evoke sessions		
Pern You ca	nissions In attach up t Search Policy n	policies (5) Info to Tommaged policies.	Туре	Filte	by Type ypes •
		mazonSageMaker-ExecutionPolicy-20240728T224074	Customer managed	1	
Ama	vers.	Maker-ExecutionPolicy-20240728T224074			
4 - 5 6 - 7 8 9 10 11 12 - 13 14 15 16 - 17 18 20	{ } {	<pre>"#Ffect: "Allow", "Action": "330et00ject", "330et00ject", "330et000ject", "330et000ject", "anniawis31::"" ] "Effect: "Allow", "Action" "sate00ject", "Action" "sate00ject", "Resource": "anniawis31::x22239243-spm/stroke_pre</pre>	diction_dataset.csv*		
	•	AmazonSageMakerCanvasAlServicesAccess	AWS managed	2.	
	•	AmazonSageMakerCanvasDataPrepFullAccess	AWS managed	1	
	•	AmazonSageMakerCanvasFullAccess	AWS managed	2	
	•	AmazonSageMakerFullAccess	AWS managed	5	

Figure 3. SageMaker execution Role Permission

### 2. Bedrock Role:

This role should have permissions to interact with Bedrock models and access the necessary AWS services.

Perm You can	issio	pus to 10 managed policies.		
				Filter by Type
Q S	earch			All types
	Poli	ky name 🕻	Type $\nabla$	Attached entitie
	+	AmazonSageMakerCanvasBedrockAccess	AWS managed	1
	⊡	InvokeModel	Customer inline	0
Invok	eMo	del		
1 - { 2 3 - 4 - 5 6 7 8 9 10 }	"v "s ]	<pre>ersion": "2012-10-17", tatement": [ {</pre>	provisioned-model/300t4abgorkj"	
	÷	s3_access_to_bedrock	Customer inline	0
	+	S3FullAccessPolicy	Customer inline	0
	⊡	S3GetObjectPolicy	Customer inline	0
S3Ge	tObj	ectPolicy		
1 - { 2 3 - 4 - 5 6 - 7 8 9 10 - 11 12 13 14 15 16 }	"v "s	<pre>tersion": "2012-10-17", tatement": [</pre>	troke-prediction-xgboost/test/test.csv", troke-prediction-xgboost/output/xgboost-2024-07-31-21-16-00-337/	. 4

Figure 4. Bedrock execution Role Permission

mazon S3 > Buckets > sagemaker-us-west-2-533267211033 > Edit bucket policy
dit bucket policy
Bucket policy
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn
Bucket ARN D am:aws:s3:::sagemaker-us-west-2-533267211033 Policy
1 V [] 2 "Version": "2012-10-17", 3 V "Statement": [
4 • {
5 "Effect": "Allow",
6 V "Principal": {
7 "AWS": "arn:aws:iam::533267211033:role/service-role/AmazonSageMaker-ExecutionRole-20240728T224074"
8 ),
9 "Action": "s3:GetObject",
10 "Resource": "arn:aws:s3:::sagemaker-us-west-2-533267211033/stroke_prediction_dataset.csv"
13 "t+tect: "Allow",
12 AW3 - griffadwsitams:5352072110551701E/SERVICE-T01E/Amazunsagemaker-canvd58edF0CKK01E-202407281224073
10 D D 11
17 * Action - [ 18 "::::::::::::::::::::::::::::::::::::
10 S-Dutholart"
21 V "Resource": [
Q 22 "arn:aws:s3:::sagemaker-us-west-2-533267211033/stroke-prediction-xgboost/test/test.csv".
Q 23 "arn:aws:s3:::sagemaker-us-west-2-533267211033/stroke-prediction-xgboost/*"
24 1
25 }
26 ]
27 6

Figure 5. S3 Buckets Policy for Bedrock access

## 2 Environmental Setup

**Python:** Make sure you have installed python. If Python is not already installed on your system, you can download and install it from the <u>official Python website</u>. Follow the instruction according to your requirements. If you have already installed it, pls check with the below code to verify the latest version is installed.

```
PS C:\Users\rajar\OneDrive\Desktop\personal\stroke-prediction> python --version
PS C:\Users\rajar\OneDrive\Desktop\personal\stroke-prediction> python --version
>>
Python 3.11.5
```

```
Figure 6. Verify Python Version
```

Libraries: Make sure to install the required python libraries as shown in fig. 7.

```
import streamlit as st
import boto3
import pandas as pd
import numpy as np
from io import StringIO
import json
import time
```

Figure 7. Import Python libraries

#### SageMaker:

1. Notebook Instance: Jupyter Notebook setup is primary to create, train and deploy model, you can use local Jupyter software to work with. As shown in fig. 8 first need to create a new notebook.

Amazon SageMaker 🗙	Amazon SageMaker > Notebook instances > Create notebook instance
Getting started	Create notebook instance Amazon SageMaker provides pre-built fully managed notebook instances that run Jupyter notebooks. The notebook instances include example code for common model training and hosting exercises. Learn more [2]
<ul> <li>Applications and IDEs</li> </ul>	Notebook instance settings
Studio	
Canvas	Notebook instance name
RStudio	Stroke Prediction
TensorBoard	Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.
Profiler	Notebook instance type
Notebooks	mLt3.medium
	Platform identifier Learn more 🗹
Admin configurations	Amazon Linux 2, Jupyter Lab 3
Domains	Additional configuration
Role manager	
Images	
Lifecycle configurations	Permissions and encryption
SageMaker dashboard Search JumpStart Foundation models Computer vision models Natural language processing models Governance HyperPod Clusters	IAM role Netbook instances require permissions to call other services including SageMaker and S3. Choose a role or let us create a role with the AmazonSageMakerFutAccess IAM policy attached.
Ground Truth     Processing	Network - optional
Training	
Inference	Git repositories - optional
Augmented Al	• Tage optional
AWS Marketplace	Tags - optional
	Cancel Create notebook instance

Figure 8. SageMaker Notebook Instance

For more user experience you can use SageMaker studio, Once Instance is created you can view the status as Running. Make sure to stop the service before closing every time.

C JupyterLab					
✓ About					
JupyterLab is the latest web-based IDE for	r notebooks, code, and data. You can selec	t your image and instance to harness the p	ower of AWS when running your machin		
See features 🖄 🛛 Quick start guide 🖉					
Q Search					
Filter spaces: Running					
Name	Application	Status	Туре		
stroke-prediction	C JupyterLab	📀 Running	Private		

Figure 9. Notebook Instance - Inservice

2. Dataset load from S3: By specifying bucket name & region we can use objects from it.



Figure 10. Dataset load from S3 to be used entirely

**3. Model Training, Endpoint Deployment:** The model training job with creating and endpoint configurations are handled in code itself. Follow the below code,



Figure 11. Model Training Job

Once The training job has been completed your model is created and ready to deploy.



Figure 12. Deploy model as Endpoint to test by invoking with prediction data.

xgboost-2024-0	8-10-22-12-53-235 (Endpoint)			
Endpoint summary				
inference Type Real-time	States © In service	Creation time Sat Aug 10 2024 23:12:54 GMT+0100 (Irish Standard Time)	Last updated Sat Aug 10 2024 23:16:20 GMT+0100 (irish Standard Time)	ARN am:aws:sagemaker:us-west- 2-533267211033:endpoint/xgboo 2024-08-10-22-12-53-235
Variants Settings Test Inf	erence Auto-scaling			
Test Inference Test your model: by sending request p Testing Options Select the method you want to use to Test the sample request Use Pythen SOK example code C Content type Test/Cor	ayload to the Sugehalar hosted endpoint and receiving a respon test your inference	ne, or by using the Tythan 3DK example code provided.		
CSV U Send the Body of InvokeEndpoint API	as text/csv format to your endpoint.		<ul> <li>Inference Result</li> </ul>	
(*************************************	, , , , , , , , , , , , , , , , , , ,		Son Access A	Envaluation Lingu 920 Result Trans Linguistic Age
		Send Request	<pre>{     "body": " 30,0,0,0,2,0,1,182.13,21.05,1,3,2,     "contentType": text/civ,     "endpointMase": "agboost-2824-08-10-22-12-53 }</pre>	0,1,2,4.16,78,167,89,77,88°, 1-235°

Figure 13. Evaluating the Endpoint by giving sample inputs

From the above figure, we can see the SageMaker endpoint evaluated successfully which has given the probability prediction score for risk of patient's stroke.

**Bedrock:** 

<ul> <li>How it works</li> </ul>			
Create a model	Purchase provisioned throughput to use the custom model		
	1		
Select a model to customize and provide the Amazon S3 bucket that contains the training (and validation, if	Purchase provisioned throughput to use a cuttom fine-tunied model, a continued pre-trained model or a copied model. You can then use your model in the Amazon Bedrock playgrounds or in your applications.		
approaches to be an state you cookin models will during accounts in you have an initial organizations of account. You can copy a model that you own or that has been shared with you to the region that you want to use it in.	поека, ток от окал ож учит телес и по иливал аконоса роу у оклас от и учи аруковното.		
approach, you don't not be suited in the procession moves and to be account in you have an india agging account of the account in the account in the second in the you own or that has been shared with you to the region that you want to use thin.  Models Jobs			
Account You can caply a model that you own or that has been shared with you to the region that you want to use thin. Models Jobs Models (1)	O     Purchase provisioned plinuighput     Cattomice model		
page and page and the formation of the page association material and the page association of page and the page association of the page associatio	Order Fine surface and a generation of the second of		

Figure 14. Bedrock custom model

As shown in fig. 14, Bedrock console has an option named custom model, here we need to select create continued pre-training job. Following this we need to select any available source model in specific region. If you have no available base model, then request for model access. Text Generation is the objective so Titan Text G1-Express need to be chosen. With the dataset stored from S3 need to select as input data. Note: Bedrock accepts only JSON format data, so the prior conversion is required from CSV file to JSON. That needs to be used as input data. The service role should be chosen appropriately as shown in fig. 15.

	Huperparameters into		
Amazon Bedrock > Custom models > Create Fine-tuning job			
Create Continued Pre-training job info	Epochs The total number of iterations of all the training data in one cycle for training the model.		
Select the model you wish to pre-train and submit your data location.	5		
	Enter an integer between 1 and 10.		
Model details	Batch size		
	The number of samples processed before model parameters are updated.		
Source model Choose from a list of models that you wish to customize with using your own data.	1		
a Titan Text G1 - Express vi	Enter a value between 1 and 64.		
Change	Learning rate		
Continued pre-trained model name	The rate at which model parameters are updated after each batch of training data.		
Enter a name to identify the newly created pre-trained model.	0.0001		
stroke-prediction	Enter a float value between 0 and 1.		
	Learning rate warmup steps		
Model encryption Info	Number of iterations over which learning rate is gradually increased to the initial rate specified.		
	5		
Tags - optional	Enter an integer between 1 and 250.		
Doer and the same to identify the training job necessary to pre-train and create a new model.     Enter a name here	Slocation     Q i3//sagemaker-as-west-2-553267211033/stroke-prediction-sigboo X View (2)     Browse 53		
► Tags - optional	Service access Info		
	Bedrock model customization job requires permissions to write to S3 on your behalf.		
VPC settings - optional			
Choose a VPC configuration to access Amazon S3 data source located in your virtual private cloud (VPC). You can create and manage VPC,	Choose a method to authorize Bedrock		
subnets and security groups in Amazon VPC 🖸	Use an existing service role Create and use a new service role		
Input data late	Service role		
Choose a file in the S3 location. The files you choose must be in the dataset format 🔀 that the model needs for	AmazonSagemakerCanvasBedrockRole-20240728T224073		
training. You can also use Sagemaker Ground Truth to create and label training datasets. Learn more 🗹			
\$3.location			
Q s3://x22239243-spm/stroke_prediction_dataset.jsonl X View 2 Browse 53	After this custom model is created, you need to purchase provisioned throughput to be able     to use this model.		
Validation dataset S3 location (optional)			
Q s3://bucket/path-to-your-data/	Cancel Create Continued Pre-training job		

Figure 15. Bedrock custom model creation

Once the training job getting completed, we need to purchase provisioned throughput as deploy and used for evaluating the custom model. As shown in fig. 16

	ut details						
Provisioned throughput name							
Stroke-Prediction							
Jame can have up to 40 characters, and it must be unique. Valid characters A-Z,a-z,0-9, and - (hyphen)							
Continued Pre-train	to create a provisioned throughput.						
► Tags - optional							
No commitment Model units I model unit delivers a specific through 1 1 Estimated purchase sum fo view the provisioned throughput	ighput level for the specified model. Model unit, ut. To request an increase, use the limit increase many pricing, visit Amazon Bedrock pricing 2,	guotas depend on the level of commitment you     form 2					
stimated hourly cost	Estimated daily cost	Estimated monthly cost					
\$20.50	\$492.00	\$14,965.00					
		Learn more					

Figure 16. Bedrock Provisioned throughput purchase.

As shown in below fig. 17, we need to evaluate the purchased custom model with sample input prompt to check the prediction as model performance.

Getting started		
Overview	a stroke-prediction	3 6
Examples	Change	
Providers		
Foundation models	Age: 27, Gender: Male, Hypertension: No, Heart Disease: Yes, Mantal Status: Married, Work Type: 2, Residence Type: Urban, Average Glacose Level: 228.09, Body Mais Index (II) Smokin Status; S. Alcohol Index Yes, Physical Activity: Inactive Strate History: No. Emitty History: Urbantis: UrbantHistor: Strote Status; 30.0 Society: 80.0 Society: 10.0	4I): 36.6, Diastolic
Base models	90, HDL: 60, LDL: 100.	
Custom models		
Imported models Preview	This is a person details, I need to check the person is in risk of stroke or not?	
Playgrounds	The person is at risk of stroke due to their high blood pressure, heart disease, and unhealthy lifestyle.	
Chat		
Text		
Image		
Builder tools		
rompt management Preview		
inowledge bases		
Igents		
Prompt flows Preview		
Safeguards		
Suardraits		
Watermark detection		
oference		
Provisioned Throughput		
Assessment		
nodel Evaluation		
terre and the F.B.		

Figure 17. Custom model performance evaluation

#### **Streamlit:**

IAM > Users > streamlit-app-user				
streamin-app-user info				
Summary				
ARN  am:aws:iam::533267211033:user/streamlit-app	0-user	Console access Disabled		Access key 1 AKIAXYK JTAMM2U42NENU Sed 3 days ago. 11 days old.
Created July 31, 2024, 00:08 (UTC+01:00)		Last console sign-in -		Access key 2 Create access key
Permissions Groups Tags Securi	ty credentials Acco	ess Advisor		
Permissions policies (3) Permissions are defined by policies attached to the user dire	ctly or through groups.			C
			Filter by Type	
Q Search			All types	•
Policy name 🖸			▲ Туре	
AmazonS3ReadOnlyAccess			AWS managed	
AmazonSageMakerFullAccess			AWS managed	
BedrockFullAccess			Customer inline	

Figure 18. Streamlit app user access permissions



Figure 19. Integration of SageMaker & Bedrock using endpoints

As fig. 18 ensures to create the IAM Users access to invoke the endpoints seamlessly. Where setting AWS credentials explicitly to get the access key. Make sure to keep the access key is in active and configured properly. Steps to customize the UI, including setting up input forms and displaying the comparison metrics (inference times, risk predictions, etc.).

```
    PS C:\Users\rajar\OneDrive\Desktop\personal\stroke-prediction> python app.py
    2024-08-11 01:29:23.866
        Warning: to view this Streamlit app on a browser, run it with the following command:
        streamlit run app.py [ARGUMENTS]
        2024-08-11 01:29:23.871 Session state does not function when running a script without `streamlit run`
        PS C:\Users\rajar\OneDrive\Desktop\personal\stroke-prediction> streamlit run app.py
        You can now view your Streamlit app in your browser.
```

Local URL: http://localhost:8502 Network URL: http://192.168.1.34:8502

Figure 20. To run Streamlit application

Using the above commands, we can test the Streamlit web UI to evaluate the stroke prediction (Text generation). Here we can to the comparison for prediction, f1 score, inference time, etc. It can be tested locally.



Figure 21. Streamlit application UI for comparison

Here we can get the result like the fig. 21 with the evaluation of risk prediction score, inference time between SageMaker & Bedrock. This how the research outcome we can see.

#### Visualizing Results from Stroke prediction model:



Figure 21. Confusion matrix

```
# Generate predicted probabilities for the positive class
y pred proba = []
for index, row in X_test.iterrows():
   result = xgb_predictor.predict(row.values)
    y_pred_proba.append(float(result)) # Directly append the float result
# Convert the list to a numpy array
y_pred_proba = np.array(y_pred_proba)
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc, roc_auc_score
# Assuming `y_test` is the true labels and `y pred_proba` are the predicted probabilities
# Compute ROC curve and AUC
fpr, tpr, thresholds = roc_curve(y_test, y_pred_proba)
roc_auc = auc(fpr, tpr)
# Plotting the ROC curv
plt.figure(figsize=(10, 7))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC)')
plt.legend(loc="lower right")
plt.show()
# Print AUC score
print(f'AUC: {roc_auc_score(y_test, y_pred_proba):.2f}')
```

Figure 22. Plotting ROC Curve

Fig. 21, 22 shows the steps followed for plotting the confusion matrix and ROC curve to visualize the accuracy of model. The model implemented in this project is evaluated by the accuracy, precision, recall and F1 score. The confusion matrix was utilized to evaluate values.

## References

Beragu, Suraj (2022) Effective use of Cloud Computing and Machine Learning Technologies for Smart Healthcare Applications. Master's thesis, Dublin, National College of Ireland.

Build generative AI applications on Amazon Bedrock — the secure, compliant, and responsible foundation | Amazon Web Services. (2024, June 29). Amazon Web Services. Available at: <u>https://aws.amazon.com/blogs/machine-learning/build-generative-ai-applications-on-amazon-bedrock-the-secure-compliant-and-responsible-foundation/</u>.

Mohajeri, M.A. (2024). Leveraging large language model for enhanced business analytics on AWS.