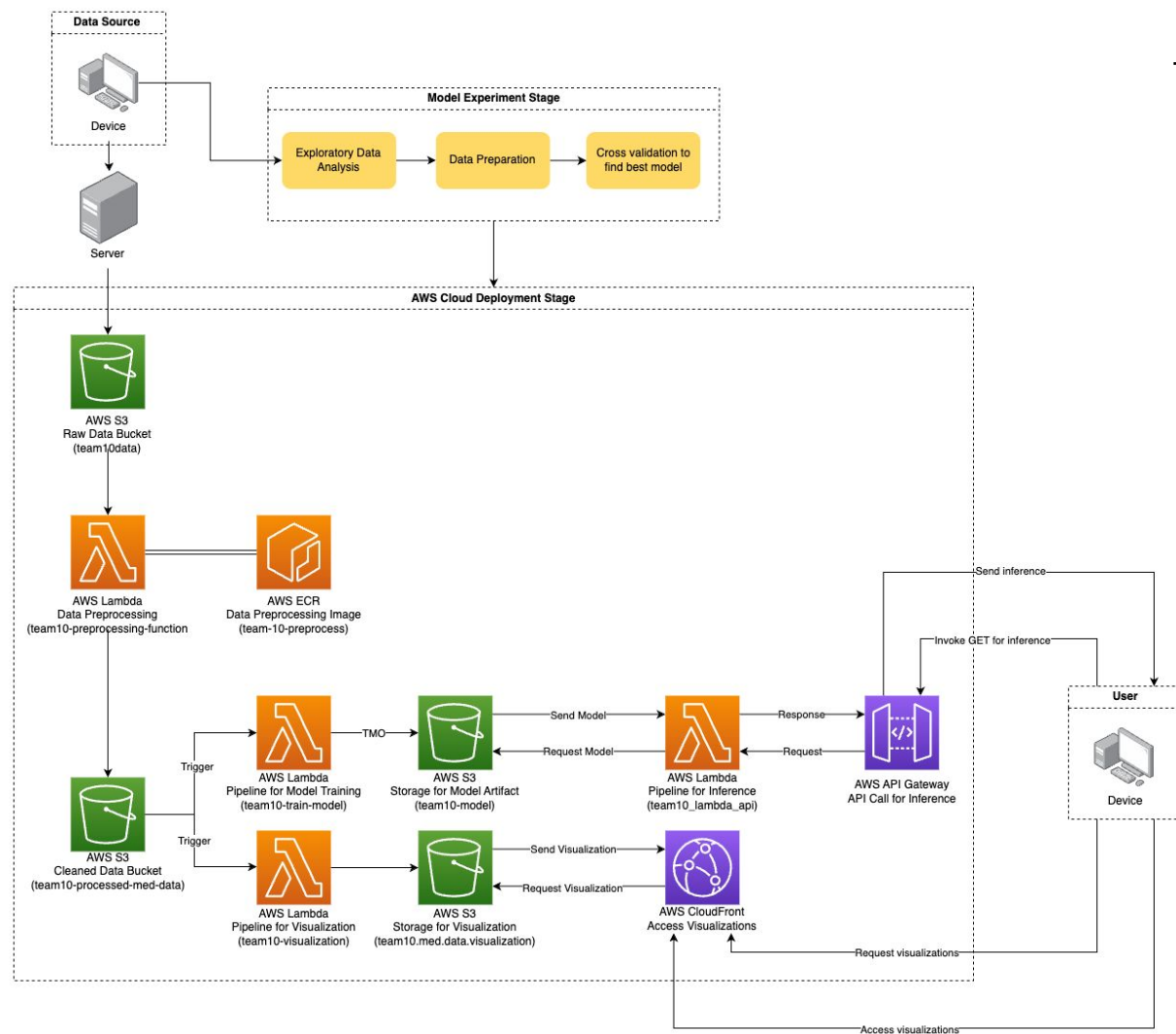


Heart Disease Data: Prediction and Visualization in AWS

MSiA 423: Cloud Engineering for Data Science Engineering
Group 10: Cindy Chen, Michelle Liu, Yifei Wang

Project Objective

- Preliminary self-check for individuals
 - Too busy or unable to seek medical resources
 - Via prediction portal
 - Preliminary estimation for heart disease for medical practitioners
 - While waiting for the results of formal medical tests
 - Via prediction portal
 - Spread understanding for heart disease patients' characteristics
 - Via visualizations
-



Total 12 months cost:
11.29 USD

Raw Data S3 Bucket

1. **Original dataset description:** The dataset is obtained from the CDC's annual survey of 400,000 adults, focuses on key indicators of heart disease, including risk factors such as high blood pressure, high cholesterol, smoking, diabetic status, obesity, physical activity, and alcohol consumption
 - 320,000 rows
2. **Triggers Processing Lambda:** Once raw data file is uploaded to s3, an event-driven Lambda is triggered to do data preparation and upsampling for the visualization and model training stage



Model Experiment Stage

1. **Exploratory Data Analysis:**

- a. Converted "Yes/No" responses related to patient disease history into numerical values (1 and 0).
- b. Converted the data types of numerical columns as needed.
- c. Plotted a correlation graph to examine the relationships between variables.
- d. Plotted feature distribution graphs to visualize the distribution of different features.

2. **Data Preparation:**

- a. Decided on the final features to be used and dropped unnecessary columns from the dataset.
- b. Upsampled the dataset to address class imbalance.

3. **Model Experimentation:**

- a. Conducted cross-validation and grid search for best hyperparameters for a random forest model.
- b. Evaluated the best model based on different metrics on the test portion of the dataset:
 - i. Accuracy: 0.77
 - ii. F1 score: 0.78
 - iii. Recall: 0.81
 - iv. Precision: 0.75



Data Preprocessing Lambda + ECR

1. **Containerized Lambda:** The Lambda function is built from a container image in private ECR, which includes the necessary Python libraries and specify the entry point for the Lambda handler. This approach ensures consistent and controlled dependencies for Lambda.
2. **Data Conversion and Transformation:**
 - a. Converts "Yes/No" responses related to patient disease history into numerical values (1 and 0).
 - b. Converts the data types of numerical columns as needed.
 - c. Drops unnecessary columns that are not required for further analysis.
 - d. Uses SMOTE to upsample the unbalanced heart disease response variable.
3. **Event-Driven S3 Trigger:** An event trigger is set up on the "team10data" S3 bucket. Whenever a raw data file is uploaded to this bucket, the Lambda function is automatically triggered. Environment variables are also included in the Lambda Configuration.
4. **Output sent to S3:** The Lambda function sends the processed data to an S3 bucket named "team10-preprocessed-med-data" for visualization and model training.



Clean Data S3 Bucket

Receives files from the team-10-preprocessing-function Lambda:

1. **Cleaned data:** This data is suitable for visualization purposes and can be used for further analysis or reporting.
2. **Upsampled data:** This data includes the balanced heart disease response and is intended for model training.

	BMI	Smoking	Stroke	PhysicalHealth	Sex	Age	Diabetic	KidneyDisease	HeartDisease
0	16.60	1	0	3.0	1	55	1	0	0
1	20.34	0	1	0.0	1	80	0	0	0
2	26.58	1	0	20.0	0	65	1	0	0
3	24.21	0	0	0.0	1	75	0	0	0
4	23.71	0	0	28.0	1	40	0	0	0



Model Training Lambda

1. Receives preprocessed data from team10-preprocessed-med-data S3 bucket
2. Performs 80-20 train-test data split
3. Fits the best **random forest classification model** obtained from model experiment stage
 - a. N_estimators = 200
 - b. Min_samples_split = 50
 - c. Min_samples_leaf = 50
 - d. Max_features = sqrt
 - e. Max_depth = None
 - f. Bootstrap = False
4. Saves the trained model artifact to team10-model S3 bucket to be used by the inference

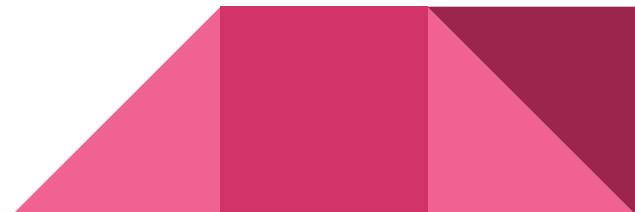


Serverless Inference with Lambda, API Gateway and Front End

Step	Spotlight
Step 1: User Interaction with Frontend Page: The user interacts with a web page by inputting their personal health data. Once the user submits their data, the webpage gathers this information and constructs an API request. This request is sent to an endpoint exposed by the AWS API Gateway using HTTP API.	<ul style="list-style-type: none">• User input error prevention• Additional question description box
Step 2: API Gateway to Lambda Invocation: AWS API Gateway receives the API request and triggers an AWS Lambda function, passing along the user data received from the frontend page.	
Step 3: Model Deserialization and Prediction: The Lambda function retrieves model pickle file from S3 and deserializes it back into a machine-learning model. The model then uses the user data to make a prediction about the likelihood of the user having heart disease.	<p>Model caching</p> <ul style="list-style-type: none">• Reduce latency• Increase scalability

Serverless Inference with Lambda, API Gateway and Front End

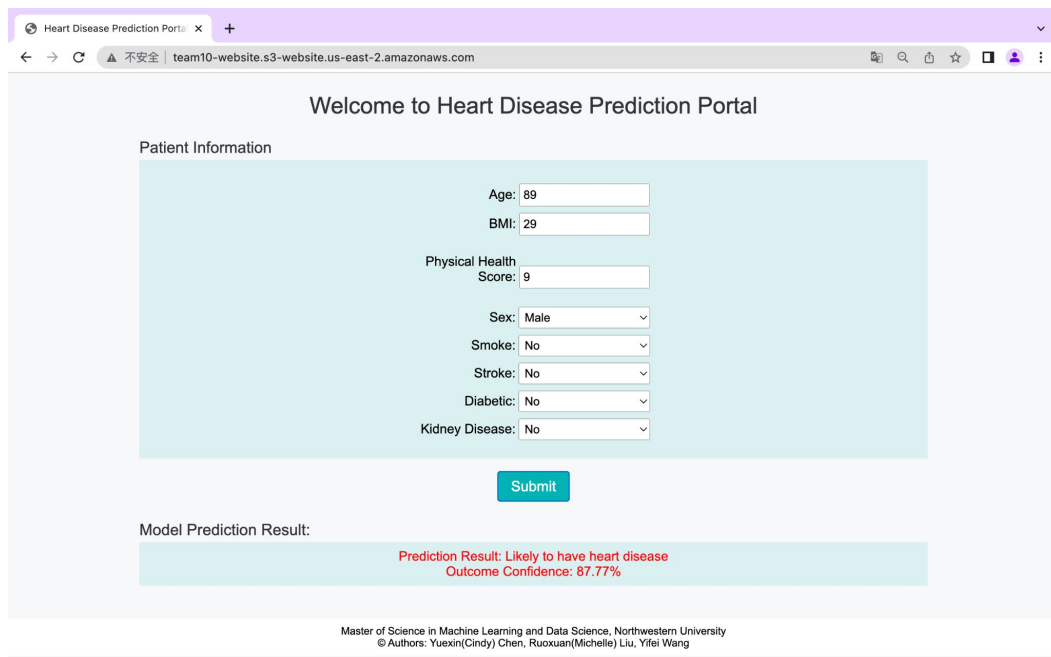
Step	Spotlight
Step 4: Response Construction and Delivery: The prediction result is then packaged into a HTTP response and sent back through the AWS API Gateway.	
Step 5: Frontend Updates: The frontend page receives this response, unpacks the prediction result, and displays it to the user.	<ul style="list-style-type: none">• Color differentiation when showing positive or negative predictions• No need to reload page for multiple prediction



Model Inference Demo

Heart Disease Prediction portal:

<http://team10-website.s3-website.us-east-2.amazonaws.com>



The screenshot shows a web browser window with the URL `team10-website.s3-website.us-east-2.amazonaws.com`. The page title is "Heart Disease Prediction Portal". The main heading is "Welcome to Heart Disease Prediction Portal". Below this is a "Patient Information" section with a light blue background. It contains several input fields: "Age" (89), "BMI" (29), "Physical Health Score" (9), "Sex" (Male), "Smoke" (No), "Stroke" (No), "Diabetic" (No), and "Kidney Disease" (No). Each field is a text input with a dropdown arrow. Below these fields is a green "Submit" button. Under the "Submit" button is a "Model Prediction Result:" section with a light blue background. It displays the text "Prediction Result: Likely to have heart disease" in red and "Outcome Confidence: 87.77%" in red. At the bottom of the page, there is a footer with the text "Master of Science in Machine Learning and Data Science, Northwestern University" and "© Authors: Yuxin(Cindy) Chen, Ruoxuan(Michelle) Liu, Yifei Wang".

Heart Disease Prediction Portal

Welcome to Heart Disease Prediction Portal

Patient Information

Age: 89

BMI: 29

Physical Health Score: 9

Sex: Male

Smoke: No

Stroke: No

Diabetic: No

Kidney Disease: No

Submit

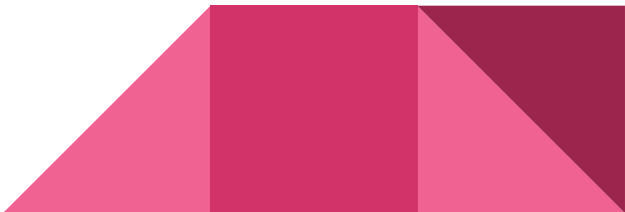
Model Prediction Result:

Prediction Result: Likely to have heart disease


Outcome Confidence: 87.77%

Master of Science in Machine Learning and Data Science, Northwestern University
© Authors: Yuxin(Cindy) Chen, Ruoxuan(Michelle) Liu, Yifei Wang

Data Visualization Lambda

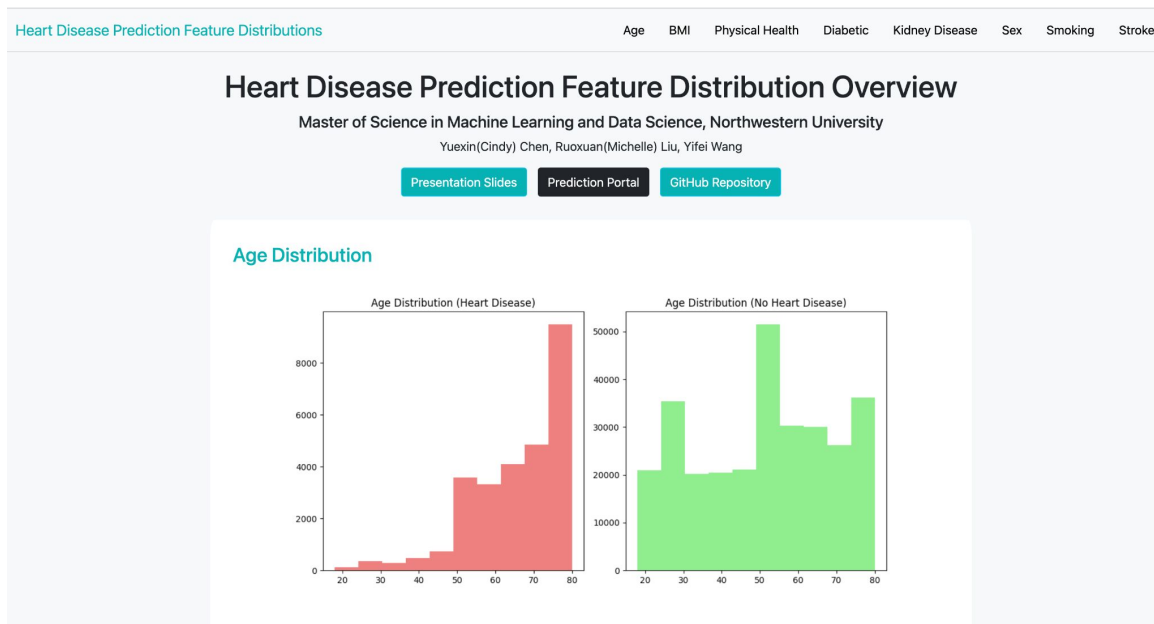
1. **Containerized Lambda:** built from a container image in private ECR "team-10-vis", includes the necessary Python libraries and specifies the entry point for the Lambda handler
 2. **Visualize Heart Disease Prediction Features:** It generates predictor distribution histograms for patients with heart disease and patients without heart disease
 3. **Input received by invoking Test:** The Lambda function is triggered by invoking the test functionality within the Lambda code section. Environment variables are used to specify the targeting S3 bucket, resulting file names, and the origin S3 bucket.
 4. **Output sent to S3:** The data visualizations generated by the Lambda function are sent to the "team10.med.data.visualization" S3 bucket.
- 

CloudFront Access for Visualization

1. **Content Delivery Network (CDN):** CloudFront acts as a global network of servers distributed across various locations worldwide.
 - a. Caching and Edge Optimization: It caches website content and delivers it to users from the server closest to their geographic location, reducing latency and improve the website's performance.
 - b. Global reach and scalability: accommodate increased demand and ensures the website remains available and responsive to users worldwide.
 2. **Use visualization S3 bucket as the origin:** CloudFront uses public "team10-preprocessed-med-data" S3 as the origin to host public visualization website.
 3. **Deliver visualization website for predictors distributions:** CloudFront protects public s3 bucket address and delivers the website content through protected link.
- 

Feature Visualization Demo

Visualization Website: <https://dzisgg772bofj.cloudfront.net/>



Relevant Links

1. [Original dataset](#)
2. [AWS price estimate](#)
3. [AWS architecture diagram](#)
4. Project GitHub repository: <https://github.com/MSIA/423-project-group10>
5. Access visualizations: <https://dzisgg772bofj.cloudfront.net/>
6. Prediction portal: <http://team10-website.s3-website.us-east-2.amazonaws.com>

