

# AMAZON SAGEMAKER AI- FROM DATA WRANGLING TO REST API

## 1. Open Amazon SageMaker AI

The screenshot shows the Amazon SageMaker AI service page. At the top, there's a search bar with 'Amazon Sagemaker AI' and a navigation bar with AWS logo, user info (Asia Pacific (Mumbai), rya @ 6953-0934-6212), and a 'Create application' button. On the left, a sidebar lists 'Services' (Features, Resources New, Documentation, Blog posts, Knowledge articles, Events, Marketplace, Tutorials) and a 'Were these results helpful?' poll ('Yes' or 'No'). The main content area is titled 'Services' and shows three cards: 'Amazon SageMaker AI' (Build, Train, and Deploy Machine Learning Models), 'AWS Lake Formation' (AWS Lake Formation makes it easy to set up a secure data lake), and 'Amazon SageMaker' (The center for data, analytics, and AI). Below this is a 'Features' section with 'SageMaker Ground Truth' (Amazon SageMaker AI feature), 'Autopilot' (Amazon SageMaker AI feature), and 'SageMaker Studio' (Amazon SageMaker AI feature). A 'Cost breakdown' section shows an 'Access denied' message. At the bottom, there's a footer with links to Privacy, Terms, and Cookie preferences.

## 2. Click on Domains under Amazon Sagemaker AI and create a domain

The screenshot shows the 'Domains' page under the Amazon SageMaker AI service. The left sidebar includes sections for Applications and IDEs (Studio, Canvas, RStudio, Notebooks, Partner AI Apps NEW), Admin configurations (Domains, Role manager, Images, Lifecycle configurations), and JumpStart (Foundation models). The main content area has a 'Domains (0)' heading with an 'Info' link. A callout box says 'Introducing domain-level resource visibility: SageMaker now allows you to view running applications, jobs and endpoints in the domain to help you monitor and manage cost. Click on the domain and go to the "Resources" tab on a domain details page.' Below this is a table header for 'Name', 'Id', 'Status', 'Created on', and 'Modified on'. A note says 'No domains' and 'To add a domain, choose Create domain.' At the bottom, there's a footer with links to Privacy, Terms, and Cookie preferences.

Click on set up for single user

The screenshot shows the 'Set up SageMaker Domain' page. On the left, there's a sidebar with navigation links like 'Getting started', 'What's new', 'Applications and IDEs' (Studio, Canvas, RStudio, Notebooks, Partner AI Apps), 'Admin configurations' (Domains, Role manager, Images, Lifecycle configurations), 'JumpStart' (Foundation models), and 'CloudShell'.

The main content area has two sections: 'Set up for single user (Quick setup)' and 'Set up for organizations'. The 'Set up for single user' section is selected, showing a list of features: New IAM role with AmazonSageMakerFullAccess policy, Public internet access, and standard encryption; SageMaker Studio - New, and SageMaker Studio Classic integrations; Sharable SageMaker Studio Notebooks; SageMaker Canvas; and IAM Authentication. A note below says 'Perfect for single user domains and first time users looking to get started with SageMaker.' The 'Set up for organizations' section lists more advanced options like Advanced network security, and data encryption; SageMaker Studio - New, SageMaker Studio Classic, RStudio, and Code Editor Based on Code-OSS, Visual Studio Code Open Source Integrations; SageMaker Studio Projects, and Jumpstart; SageMaker Canvas, and Amazon services integrations; and IAM, or IAM Identity Center (successor to AWS SSO). A note below says 'Better for admins with large user groups, but you can always update your account configuration settings later if you want to do a quick setup now.'

At the bottom right of the main content area is a yellow 'Set up' button. At the very bottom of the page are footer links: © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

The screenshot shows the 'Domain: QuickSetupDomain-20250501T153207' page. At the top, there's a green banner with the message 'The SageMaker Domain is ready. Choose your user name, then choose Launch app to get started.' Below this, the page title is 'QuickSetupDomain-20250501T153207' and the sub-section is 'Domain details'. There are tabs for 'Domain settings' (selected), 'User profiles', 'Space management', 'App Configurations', 'Environment', and 'Resources'. Under 'Domain settings', there's a 'General settings' section with fields for Name (QuickSetupDomain-20250501T153207), Status (Ready), Domain ID (d-zzign3umvng0), Created (Thu May 01 2025 15:32:26 GMT+0530 (India Standard Time)), and Last modified (Thu May 01 2025 15:35:16 GMT+0530 (India Standard Time)). Below this is a 'Domain rules' section with a table header 'Rule type', 'Application type', 'Rule action', and 'Resource'. The table body shows 'No domain rules' and 'No domain rules to display'. At the bottom of the page are footer links: © 2025, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

### 3.Create a user in the domain

The screenshot shows the Amazon SageMaker AI console interface. On the left, a sidebar navigation includes sections for Getting started, What's new, Applications and IDEs (Studio, Canvas, RStudio, Notebooks, Partner AI Apps), Admin configurations (Domains, Role manager, Images, Lifecycle configurations), and JumpStart (Foundation models, CloudShell, Feedback). The main content area displays the 'Domain details' for 'QuickSetupDomain-20250501T153207'. The 'User profiles' tab is selected, showing a table with one row: 'default-20250501T153207' created on May 01, 2025, at 10:05 UTC. A 'Launch' button is visible next to the row. A success message 'User profile was successfully created.' is displayed in a green bar at the top of the page.

4. After creating the domain and user go to studio under Amazon Sagemaker AI and open studio with the user profile we created in the domain.

The screenshot shows the Amazon SageMaker AI Studio interface. On the left, there's a sidebar with navigation links: 'Getting started', 'What's new', 'Applications and IDEs' (with 'Studio' selected), 'Admin configurations', 'JumpStart', and 'JumpStart' again. The main content area features a large heading 'SageMaker Studio' with the subtext 'The first fully integrated development environment (IDE) for machine learning.' Below this is a 'Get Started' box with a dropdown for 'Select user profile' set to 'default-1746094278090' and a 'Open Studio' button. Further down, there's a 'What's new' section with a card about reusing project names and a 'Pricing (US)' section with information about pay-as-you-go pricing.

5. After opening Studio click on Data wrangler and open it.

The screenshot shows the SageMaker Studio Home page. The sidebar on the left has a 'Data' section with 'Data Wrangler' highlighted. The main content area features an 'Onboarding plan' box with three items: 'Take the tour', 'Access your EFS data in JupyterLab and CodeEditor', and 'Access your Studio Classic apps'. Below this is a 'Prepare, transform and store data' section and an 'Overview' section. At the bottom, there are two large buttons: one orange for 'jupyter' and one blue for 'code'.

## Click on Run in Canvas

The screenshot shows the SageMaker Studio interface for the Data Wrangler application. The left sidebar has a 'Data' section expanded, showing 'Data Wrangler' under 'Aggregate & prepare data'. The main content area is titled 'Data Wrangler' and describes it as a tool for building Generative AI or ML solutions in SageMaker Canvas without code. It features a 'Run in Canvas' button which is currently labeled 'Stopped'. Below this, there's a section titled 'How Data Wrangler works' with four categories: Import data, Prepare data, Export, and Train. Under 'Examples and tutorials', there are four cards: Tabular data, Image data, Text data, and Time series data.

## Click on Open in Canvas

This screenshot is identical to the one above, showing the SageMaker Studio Data Wrangler interface. The key difference is the status of the 'Run in Canvas' button, which is now labeled 'Running' with a green checkmark. A green banner at the bottom of the screen also states 'Canvas is now running'.

6.Click on Data wrangler and then import and prepare to import data needed.

The screenshot shows the Data Wrangler interface. On the left, a sidebar lists various services: Home, Amazon Q, Data Wrangler (selected), Datasets, My Models, ML Ops, Ready-to-use, Gen AI, Help, and Log out. The main area is titled "Data Wrangler" and has tabs for "Data flows" and "Jobs". A flow diagram consists of four icons: "Import data" (two databases), "Prepare data" (a lightbulb), "Scale data operations" (two gears), and "Build models" (a wrench and a gear). Below the diagram, a message says "You haven't imported and prepared any data". A callout box over the "Import and prepare" button indicates "Dataset type" is selected, showing options for "Tabular" (CSV or Parquet files) and "Image" (PNG and JPG files). Other sections include "Resources" with links to "Get started" (What is Amazon SageMaker Data Wrangler, Get started with Data Wrangler in Canvas) and "Documentation" (Pricing in Canvas, AWS technical guide). A "What's new" section highlights "Accelerate data preparation for ML with comprehensive data preparation capabilities and a natural language interface".

Select the data source type and upload it

This screenshot shows the "Import tabular data" step of the wizard. It starts with a dropdown menu "Select a data source:" set to "Local upload". Below is a large dashed box for dragging files, with the placeholder "Drag a CSV or Parquet file here" and an "or" link to "Select files from your computer". To the right, a preview panel shows "1 file ready to import" with "bankloan.csv" listed, accompanied by a checkmark icon and a "Delete all" link. At the bottom are "Cancel" and "Next" buttons.

## Click on import data

Import tabular data

← Previewing 1 file Showing the first 100 rows Import settings

If your data has special character delimiters, use the advanced import settings to specify a custom delimiter. [Learn More](#)

bankloan.csv Delete

ID	Age	Experience	Income	Zi
1	25	1	49	91
2	45	19	34	90
3	39	15	11	94
4	35	9	100	94
5	35	8	45	91
6	37	13	29	92

Import settings

Settings apply to all imported files. [Learn more](#)

Dataset name \*

bankloan.csv

Sampling

Sample your dataset for faster exploration. Your full dataset will be used for data export or model build. [Learn more](#)

Sampling method \* ⓘ

Random

Random sampling ensures that each row has an equal probability of being chosen.

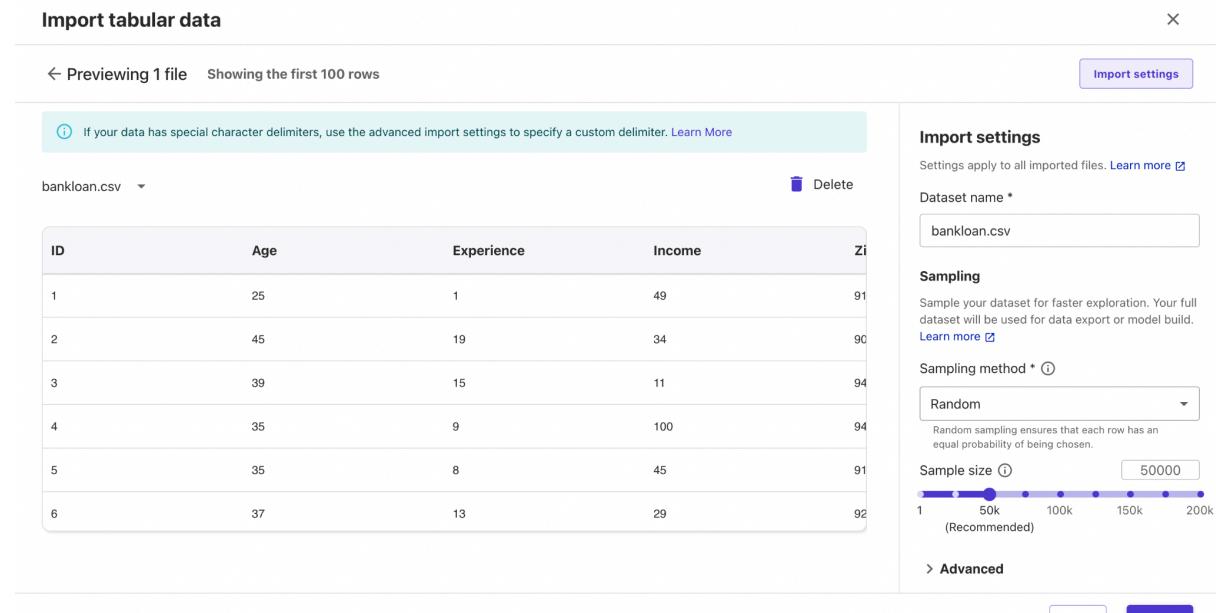
Sample size ⓘ

50000

1 50k 100k 150k 200k (Recommended)

> Advanced

Cancel Back Import



8.After Importing the Data it will appear as a flow- click on get data insights to create a Data Quality and Insights report.

Data Wrangler: Data flow > New data flow 2025-5-1 4:05:03 PM.flow Add data

Step 2: Data types Data flow Data Analyses

Validation complete 0 errors Done Run validation

+ Add transform

Get data insights Identify data quality issues and get recommendations.

Combine data

Create model Export data and start model building in Canvas.

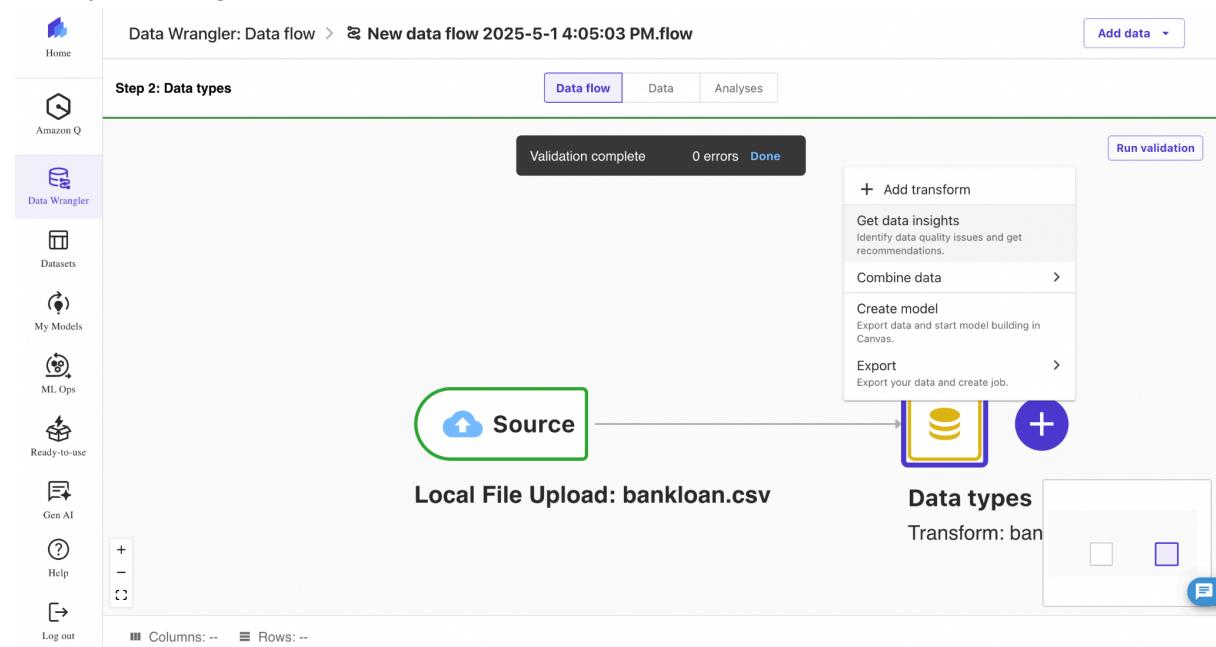
Export Export your data and create job.

Source Local File Upload: bankloan.csv

+

Data types Transform: ban

Columns: -- Rows: --



## Click create

9. After Generating the report- go back to the data flow and click on add transform- this will allow you to apply any data preprocessing steps which are needed.

Data Wrangler: Data flow > New data flow 2025-5-1 14:05:03 PM.flow > bankloan.csv

**Step 3. Impute** Chat for data prep Data flow Data Analyses Create model

Build a data insights report to identify data quality issues, recommends fixes, and estimates feature importance and model accuracy. Get data insights X

ID (long)	Age (long)	Experience (long)	Income (long)
1 - 5000	23 - 67	-3 - 43	8 - 224
1	25	1	49
2	45	19	34
3	39	15	11
4	35	9	100
5	35	8	45
6	37	13	29
7	53	27	72
8	50	24	22

Sampling: 50,000 Columns: 14 Rows: 5,000 Show visualizations

**Steps**

- + Add transform
- 1. Local File Upload: bankloan.csv
- 2. Data types
- 3. Impute

Replace, drop, or add indicators for missing values. [Learn more.](#)

Transform \* ⓘ Fill missing

Input columns \* ID x Age x Experience x Income x

Data Wrangler: Data flow > New data flow 2025-5-1 14:05:03 PM.flow > bankloan.csv

**Step 3. Fill missing** Chat for data prep Data flow Data Analyses Create model

Build a data insights report to identify data quality issues, recommends fixes, and estimates feature importance and model accuracy. Get data insights X

ID (long)	Age (long)	Experience (long)	Income (long)
1 - 5000	23 - 67	-3 - 43	8 - 224
1	25	1	49
2	45	19	34
3	39	15	11
4	35	9	100
5	35	8	45
6	37	13	29
7	53	27	72
8	50	24	22

Sampling: 50,000 Columns: 14 Rows: 5,000 Show visualizations

**← Encode categorical**

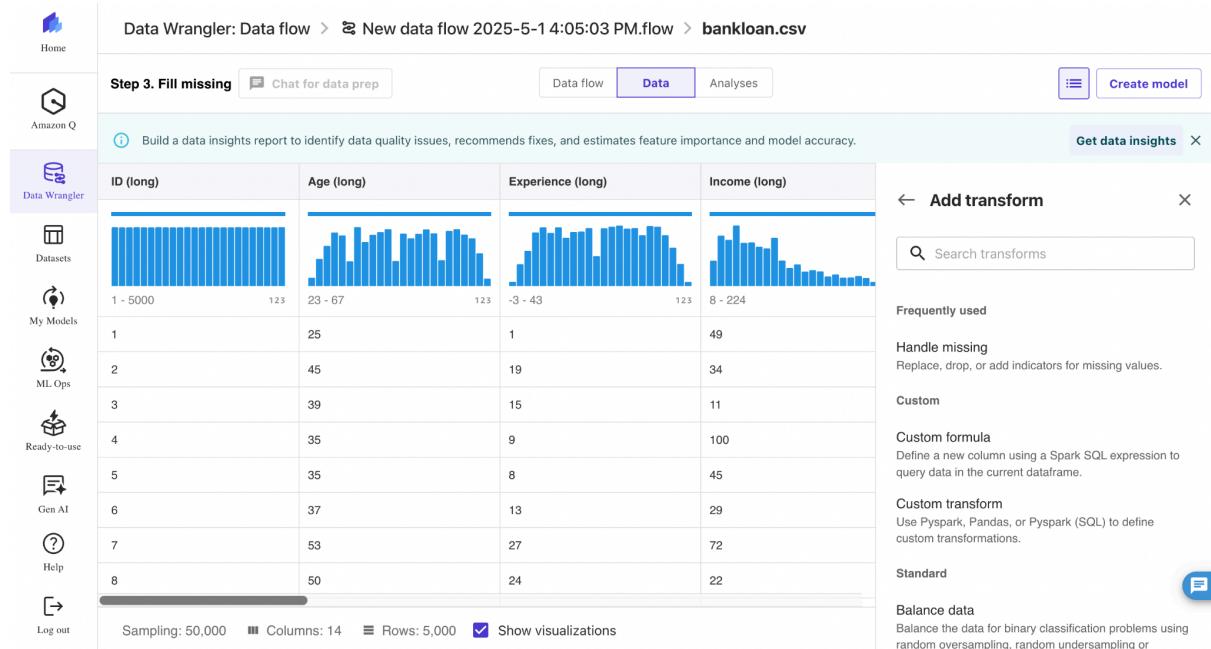
Convert categorical variables to numeric or vector representations. [Learn more.](#)

Transform \* ⓘ Ordinal encode

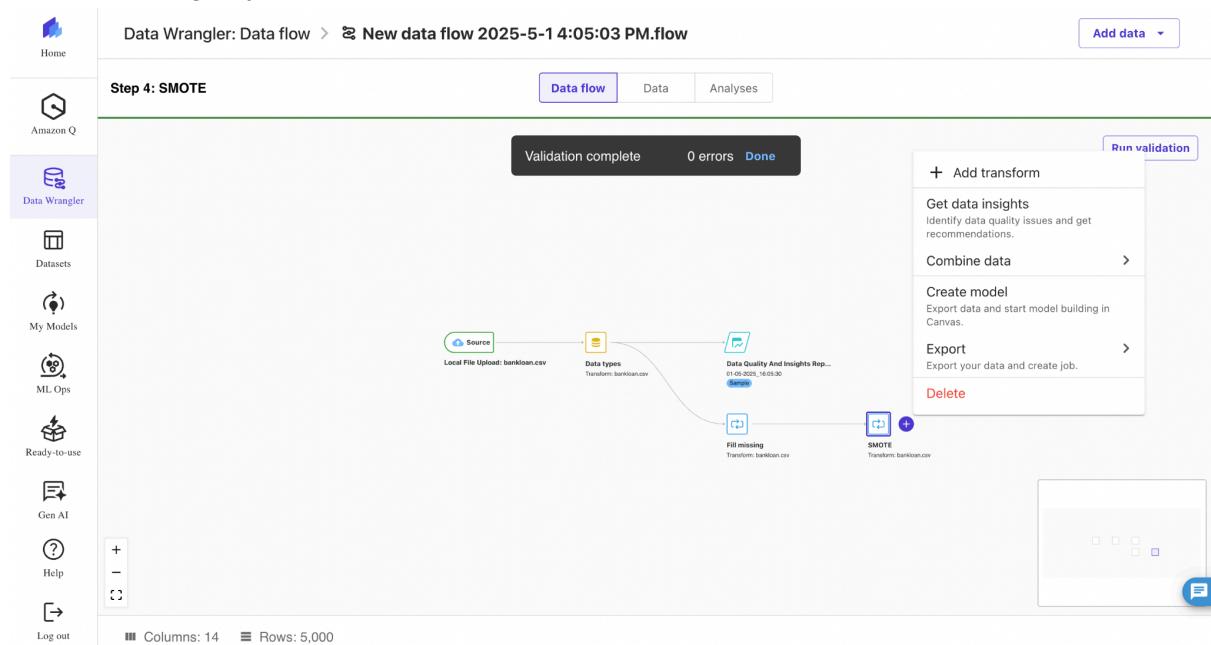
Input columns \* ⓘ Select...

**Advanced**

Clear Preview Add



## 10. After adding any data transformation steps which are required click on create model



## Click on export and create model

Data Wrangler: Data flow > **New data flow 2025-5-1 4:05:03 PM.flow**

**Step 4: SMOTE**

Validation complete    0 errors    Done

**Export to create a model**

**Dataset details**  
Dataset name \* **Dataset\_20250501\_104001**  
Use only letters, numbers, spaces, dashes, colons, and underscores up to 64 characters.

Process entire dataset ⓘ

**Model details**  
Model name \* **Model\_20250501\_104001**

Problem type \* Predictive analysis

Target column \* **Loanapproved**

**Cancel** **Export and create model**

**Run validation**

Local File Upload: bankloan.csv → Data types → Data Quality And Insights Report → Fill missing → SMOTE → Destination

Source → Data types → Data Quality And Insights Report → Fill missing → SMOTE → Destination

Running data flow on dataset...

Columns: 14    Rows: 5,000

Click on configure model to choose the model type, objective metric, training method and algorithms, Data Split and Max candidates and runtime

My models > Model\_20250501\_104001 > Version 1

Select Build Analyze Predict Deploy

**Select a column to predict**

Choose the target column. The model that you build predicts values for the column that you select.

Target column: Loanapproved

Value distribution:

**Model type**

SageMaker Canvas automatically recommends the appropriate model type for your analysis.

2 category prediction

Your model classifies Loanapproved into two categories.

**Configure model**

**Quick build**

**Preview model**

**Dataset\_20250501\_104001**

Full dataset: 9.0k rows

Column name	Data type	Feature type	Missing	Mismatched	Unique	Mode
Zipcode	123 Numeric	-	0.00% (0)	0.00% (0)	4225	94,720
SecuritiesAccount	123 Numeric	-	0.00% (0)	0.00% (0)	857	0
Online	123 Numeric	-	0.00% (0)	0.00% (0)	1935	1
Mortgage	123 Numeric	-	0.00% (0)	0.00% (0)	2285	0

Total columns: 14 Total rows: 9,040 Total cells: 126,560 Show dropped columns

My models > Model\_20250501\_104001 > Version 1

**Configure model** Reset to default settings

**Basic**

**Model type**

**Advanced - Optional**

**Objective metric**

**Training method and algorithms**

**Data split**

Specify how you want to split your data into training and validation sets. Canvas builds your model with the training set and verifies the model's accuracy with the validation set.

Configuring the data split will default to Standard build.

Training set: 80 %

Validation set: 20 %

**Max candidates and runtime**

**Cancel** **Save**

Total columns: 14 Total rows: 9,040 Total cells: 126,560 Show dropped columns

## After configuring the model click on quick build

My models > Model\_20250501\_104001 > Version 1

Select Build Analyze Predict Deploy

**Select a column to predict**

Choose the target column. The model that you build predicts values for the column that you select.

Target column: Loanapproved

Value distribution:

**Model type**

SageMaker Canvas automatically recommends the appropriate model type for your analysis.

2 category prediction

Your model classifies Loanapproved into two categories.

Configure model

**Quick build**

**Preview model**

**Dataset\_20250501\_104001**

Full dataset: 9.0k rows

Column name	Data type	Feature type	Missing	Mismatched	Unique	Mode
Zipcode	123 Numeric	-	0.00% (0)	0.00% (0)	4225	94,720
SecuritiesAccount	123 Numeric	-	0.00% (0)	0.00% (0)	857	0
Online	123 Numeric	-	0.00% (0)	0.00% (0)	1935	1
Mortgage	123 Numeric	-	0.00% (0)	0.00% (0)	2285	0

Total columns: 14 Total rows: 9,040 Total cells: 126,560 Show dropped columns

My models > Model\_20250501\_104001 > Version 1

Select Build Analyze Predict Deploy

**Preprocessing your dataset. This can take a few minutes. You can navigate away from this page. This won't interrupt the process. Learn more**

**Select a column to predict**

Choose the target column. The model that you build predicts values for the column that you select.

Target column: Loanapproved

Value distribution:

**Model type**

SageMaker Canvas automatically recommends the appropriate model type for your analysis.

2 category prediction

Your model classifies Loanapproved into two categories.

Configure model

Validating your data...

Validating your data...

**Dataset\_20250501\_104001**

Full dataset: 9.0k rows

Column name	Data type	Feature type	Missing	Mismatched	Unique	Mode
Zipcode	Numeric	-	0.00% (0)	0.00% (0)	4225	94,720
SecuritiesAccount	Numeric	-	0.00% (0)	0.00% (0)	857	0
Online	Numeric	-	0.00% (0)	0.00% (0)	1935	1

Total columns: 14 Total rows: 9,040 Total cells: 126,560 Show dropped columns

11. After the model has been created you can click on predict to test the models prediction on a dataset

My models > Model\_20250501\_104001 > Version 1

+ Create new version ⚙️ ⏪ ⋮

Select Build Analyze Predict Deploy

Model status (Quick build)

Accuracy ⓘ F1 ⓘ Optimization metric

99.06% 0.991

The model predicts the correct Loanapproved 99.06% of the time. ⓘ

Start making Predictions with your model

Predict Standard build Deploy

Overview Scoring Advanced metrics Model leaderboard

Column impact ⓘ

Search columns...

Rank	Feature	Impact (%)
1	Income	32.329%
2	Online	20.178%
3	CCAvg	13.793%
4	CDAccount	13.325%

Impact of Income on prediction of Loanapproved

Impact on prediction

Income

Dataset\_20250501\_104001 Total columns: 14 Total rows: 9,040 Total cells: 126,560 Loanapproved 2 category prediction

Predict

The screenshot shows the Amazon SageMaker Studio interface with the 'Predict' tab selected. The top navigation bar includes 'My models > Model\_20250501\_104001 > Version 1' and a '+ Create new version' button. Below the navigation is a horizontal menu with tabs: 'Select', 'Build', 'Analyze', 'Predict' (which is highlighted in blue), and 'Deploy'. The main content area is titled 'Predict target values' and contains a 'Batch prediction' button (which is highlighted in blue) and a 'Single prediction' button. A sub-section titled 'Generate predictions for an entire dataset.' includes 'Manual' and 'Automatic' buttons. Below this is a 'Predictions' section with 'All Jobs' selected (highlighted in blue) and a 'Configuration' tab. A 'Filter by configuration name:' dropdown is set to 'All'. To the right is a 'Send to Amazon QuickSight' checkbox. At the bottom is a table with columns: Job name, Created, Input dataset, Prediction type, Configuration name, Rows, and QuickSight. The 'Job name' column has a checkbox header. A large circular icon with a grid pattern is positioned in the center of the page.

## Select dataset for predictions

X

To make predictions on a dataset, select it or import it. The dataset that you select must have the same number of feature columns as the training dataset. [?](#)

[+ Create dataset](#)

Search datasets in Canvas

Name	Columns	Rows	Cells	Created	Status
<input checked="" type="radio"/> <a href="#">Dataset_20250501_104001</a>	V1	14	9,040	126,560	05/01/2025 4:10 PM
<input type="radio"/> <a href="#">canvas-sample-housing.csv</a>	V1	10	1,000	10,000	05/01/2025 4:02 PM
<input type="radio"/> <a href="#">canvas-sample-shipping-logs.csv</a>	V1	12	1,000	12,000	05/01/2025 4:02 PM
<input type="radio"/> <a href="#">canvas-sample-product-descriptions.csv</a>	V1	5	120	600	05/01/2025 4:02 PM
<input type="radio"/> <a href="#">canvas-sample-loans-part-2.csv</a>	V1	5	1,000	5,000	05/01/2025 4:02 PM
<input type="radio"/> <a href="#">canvas-sample-loans-part-1.csv</a>	V1	19	1,000	19,000	05/01/2025 4:02 PM
<input type="radio"/> <a href="#">canvas-sample-diabetic-readmission.csv</a>	V1	16	1,000	16,000	05/01/2025 4:02 PM
<input type="radio"/> <a href="#">canvas-sample-retail-electronics-forecasting.csv</a>	V1	6	40,500	243,000	05/01/2025 4:02 PM

batchInfer-Model\_20250501\_104001-Dataset\_20250501\_104001-1746096461 predictions ready [View](#) X

[Close](#)

[Generate predictions](#)

My models > Model\_20250501\_104001 > Version 1

+ Create new version

batchInfer-Model\_20250501\_104001-Dataset\_20250501\_104001-1746096461

Prediction (Loanapproved)	Probability	ID	Age	Experience	Income	Zipcode	Family
0	99.9%	1	25	1	49	91107	4
0	99.9%	2	45	19	34	90089	3
0	99.9%	3	39	15	11	94720	1
0	96.5%	4	35	9	100	94112	1
0	99.9%	5	35	8	45	91330	4
0	99.9%	6	37	13	29	92121	4
0	99.9%	7	53	27	72	91711	2
0	99.9%	8	50	24	22	93943	1
0	99.9%	9	35	10	81	90089	3

[Send to Amazon QuickSight](#) [Download](#)

batchInfer-Model\_20250501\_104001-Dataset\_20250501\_104001-1746096461 predictions ready [View](#) X

## 12. Deploying the model- Go to My Models and select view on model you want to deploy

The screenshot shows the 'My models' section of a machine learning platform. On the left is a sidebar with icons for Home, Amazon Q, Data Wrangler, Datasets, My Models (which is selected and highlighted in blue), ML Ops, Ready-to-use, Gen AI, Help, and Log out. The main area has tabs for 'Grid' and 'List'. A search bar at the top right says 'Search models' and a button says '+ New model'. Below the tabs is a filter for 'problem type: 2 category prediction'. A large card displays a summary for 'Model\_20250501\_104001': it is 'Ready' (indicated by a green checkmark). It has '1' version, 'Loanapproved' as the target, '2 category prediction' as the problem type, and was updated on '2025-5-1 4:15:06 PM'. There are 'View' and more options buttons at the bottom of the card.

Click on the model again

The screenshot shows the 'Versions' page for the model 'Model\_20250501\_104001'. The sidebar on the left is identical to the previous screenshot. The main title is 'My models > Model\_20250501\_104001'. A 'Create new version' button is at the top right. Below it is a 'Show advanced metrics' toggle switch. The 'Versions' table has columns: Version, Status, Build type, Created, Dataset, Accuracy, and Model Registry. One row is visible: Version V1, Status Ready (green checkmark), Build type Quick, Created 05/01/2025 4:10 PM, Dataset 'Dataset\_...', Accuracy 99.06%, and Model Registry Not Registered. There is a more options button at the end of the row.

## Click on deploy and create deployment

**My models > Model\_20250501\_104001 > Version 1**

Select Build **Analyze** Predict Deploy

**Model status** (Quick build)

Accuracy ( Optimization metric) F1 ( Optimization metric)  
99.06% 0.991

The model predicts the correct Loanapproved 99.06% of the time.

**Predict** **Standard build** **Deploy**

**Overview** Scoring Advanced metrics

**Column impact** (↓)

Column	Impact on prediction
1 Income	32.329%
2 Online	20.178%
3 CCAvg	13.793%
4 CDAccount	13.325%

**Impact of Income on prediction of Loanapproved**

Impact on prediction

Income

Dataset\_20250501\_104001 Total columns: 14 Total rows: 9,040 Total cells: 126,560 Loanapproved 2 category prediction

**Create Deployment**

Deploy your model to a SageMaker endpoint so that you can make predictions from outside of the Canvas application, test and monitor your model to proactively detect issues such as model drift.

**Selected model version**  
Model\_20250501\_104001  
v1 Ready Created: 05-02-2025-2:53 PM

**Deployment type**  
Real-time

**Deployment name**  
Deployment name  
new-deployment-05-02-2025-2-53-PM

**Instance type** ml.t2.medium

**Instance count**

13- To create an endpoint- we go back to SageMaker AI studio and select endpoints from deployments

The screenshot shows the SageMaker Studio Home page. On the left, a sidebar menu includes Data, Auto ML, Experiments, Jobs, Pipelines, Models, JumpStart, Deployments (selected), and Projects. Under Deployments, there are sub-options for Endpoints (Manage deployed models) and Projects (Automate model building & deployment). A large central panel displays the "Onboarding plan" with three sections: "Take the tour" (Quick tour highlights where you can find key features and how to navigate the new experience. See what's new and where to locate the tools you need to be productive.), "Access your EFS data in JupyterLab and CodeEditor" (Automatically available in private spaces.), and "Access your Studio Classic apps" (Pickup where you left off and access your Studio Classic apps from within the updated Studio experience.). Below the onboarding plan, there is a "Deploy models for inference" section with a link to "Revert to Studio Classic experience in domain settings. Learn more". At the bottom, there are tabs for Overview, Getting started, and What's new, followed by a "jupyter" logo.

Here click on create endpoint

The screenshot shows the SageMaker Studio Endpoints page. The sidebar menu is identical to the Home page, with Deployments selected. The main content area is titled "Endpoints" and contains a table of existing endpoints. The table has columns for Name, Status, Created on, and Modified on. One row is visible, showing "canvas-new-deployment-05-02-2..." with "In service" status, created on "02/05/2025, 14:54:10" and modified on "02/05/2025, 15:02:05". Below the table, there is a "Learn about endpoints" section with links for Get started, Documentation, and What's new. At the bottom, there are links for Privacy, Site Terms, and Cookie Preferences, along with a copyright notice: "© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved."

Click on add model and select the ML model we created

The screenshot shows the 'Create endpoint' settings page in AWS SageMaker Studio. On the left, a sidebar navigation includes 'Data', 'Auto ML', 'Experiments', 'Jobs', 'Pipelines', 'Models', 'JumpStart', 'Deployments' (selected), 'Endpoints' (under Deployments), and 'Projects'. The main area is titled 'Create endpoint' with the sub-section 'Endpoint settings'. It contains fields for 'Endpoint name' (set to 'Endpoint-20250502-100134'), 'Instance type' (set to 'ml.c6i.xlarge'), 'Initial instance count' (set to '1'), and 'Maximum instance count' (set to '20'). Under 'Inference type', 'Real-time' is selected, with a note: 'For sustained traffic and consistently low latency. Supports payload sizes up to 6 MB and runtimes up to 60 sec.' A 'Models' section has a '+ Add model' button. At the bottom right are 'Cancel', 'Deploy' (disabled), and a message bubble icon.

Choose deployable model as we have created the model on AWS Sagemaker

The screenshot shows the 'Add model' dialog in AWS SageMaker Studio. The sidebar on the left is identical to the previous screenshot. The dialog title is 'Add model' and the sub-section is 'Step 1: Select a model from'. It offers two options: 'JumpStart Foundation Models' (radio button not selected) and 'Deployable Models' (radio button selected). A note below states: 'Models with network isolation, IAM role, or VPC config inconsistent with the endpoint configuration are disabled. Other incompatible models are also disabled'. A search bar 'Search by model name' is present. A table lists five models, all of which are currently 'None' in the 'Status' column. The table has columns: Name, Created On, Deployed Endpoints, and Status. The first model listed is 'canvas-model-2025-05-02-09-...'. At the bottom are 'Cancel', '+ Add model' (disabled), and 'Deploy' (disabled).

Name	Created On	Deployed Endpoints	Status
canvas-model-2025-05-02-09-...	02/05/2025, 14:54:09	1	None
canvas-model-2025-05-01-11-...	01/05/2025, 16:52:20	0	None
canvas-model-2025-05-01-10-...	01/05/2025, 16:20:50	0	None
canvas-model-2025-05-01-10-...	01/05/2025, 16:20:32	0	None
canvas-model-2025-05-01-10-...	01/05/2025, 16:20:08	0	None

## Create the endpoint

The screenshot shows the SageMaker Studio interface for creating a new endpoint. The left sidebar is visible with various navigation options like Data, Auto ML, Experiments, etc. The main area is titled "Endpoint-20250502-100134" and displays the "Endpoint summary". Key details shown include:

- Inference Type: Real-time
- Status: Creating
- Last updated: Fri May 02 2025 15:33:19 GMT+0530 (India Standard Time)
- ARN: arn:aws:sagemaker:ap-south-1:695309346212:endpoint/Endpoint-t-20250502-100134
- Creation time: Fri May 02 2025 15:33:19 GMT+0530 (India Standard Time)
- URL: https://runtime.sagemaker.ap-south-1.amazonaws.com/endpoints/Endpoint-20250502-100134/invocations
- Endpoint logs: /aws/sagemaker/endpoints/Endpoint-20250502-100134

Below the summary, there are tabs for Models, Settings, and Test inference. A success message is displayed: "Endpoint Endpoint-20250502-100134 is being created." and "Successfully add inference component: canvas-model-2025-05-02-09-24-07-581714-20250502-1002510". At the bottom, there are buttons for Delete, Add model, and a message icon.

To test the endpoint click on  
Test inference,  
choose text/csv  
paste- 1,24,4,300000,400053,1,400,4,0,1,1,1,0

The screenshot shows the "Test inference" tab in SageMaker Studio. The left sidebar is visible. The main area displays the "Inference Result" for a successful request. Key details shown include:

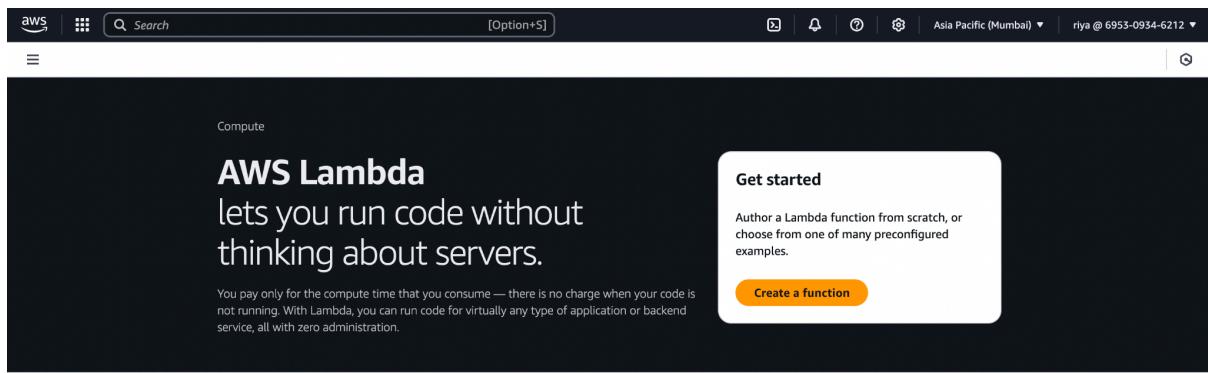
- Status: Success
- Execution Length (ms): 259
- Request Time: 1 minutes ago
- Result Time: 1 minutes ago
- Result content (JSON):

```
{  "body": {},  "contentType": "text/csv",  "invokedProductionVariant": "variant-1"}
```

A "Copy entire result" button is located at the bottom right of the result panel.

14-After the endpoint has been created we will make a Rest API-

Open AWS Lambda and click on create a function

This screenshot shows the 'How it works' section of the AWS Lambda service. It includes tabs for '.NET', 'Java', 'Node.js', 'Python' (which is selected), 'Ruby', and 'Custom runtime'. Below the tabs is a code editor window containing Python code for a lambda handler. The code is as follows:

```
1 def lambda_handler(event, context):
2     print(event)
3     return 'Hello from Lambda!'
4
```

A 'Run' button is located at the top right of the code editor. A tooltip above the 'Run' button says 'Next: Lambda responds to events'. At the bottom of the page, there are links for 'CloudShell', 'Feedback', '© 2025, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

Scroll down to code source and put the following code in and click deploy-

```
import boto3
import json

runtime = boto3.client("sagemaker-runtime")

def lambda_handler(event, context):
    try:
        body = event["body"]
        input_json = body if isinstance(body, dict) else json.loads(body)

        feature_order = [
            "ID", "Age", "Experience", "Income", "Zipcode", "Family",
            "CCAvg", "Education", "Mortgage", "SecuritiesAccount",
            "CDAccount", "Online", "CreditCard"
        ]

        csv_values = [str(input_json[feature]) for feature in feature_order]
        csv_string = ",".join(csv_values)

        response = runtime.invoke_endpoint(
            EndpointName="Endpoint-20250502-100134",
            ContentType="text/csv",
            Body=csv_string.encode("utf-8"),
            Accept="application/json",
        )
```

```

InferenceComponentName="canvas-model-2025-05-02-09-24-07-581714-20250502-1002510"
)

result = response["Body"].read().decode("utf-8")

return {
    "statusCode": 200,
    "headers": {"Content-Type": "application/json"},
    "body": result
}

except Exception as e:
    return {
        "statusCode": 500,
        "body": json.dumps({"error": str(e)})
}

```

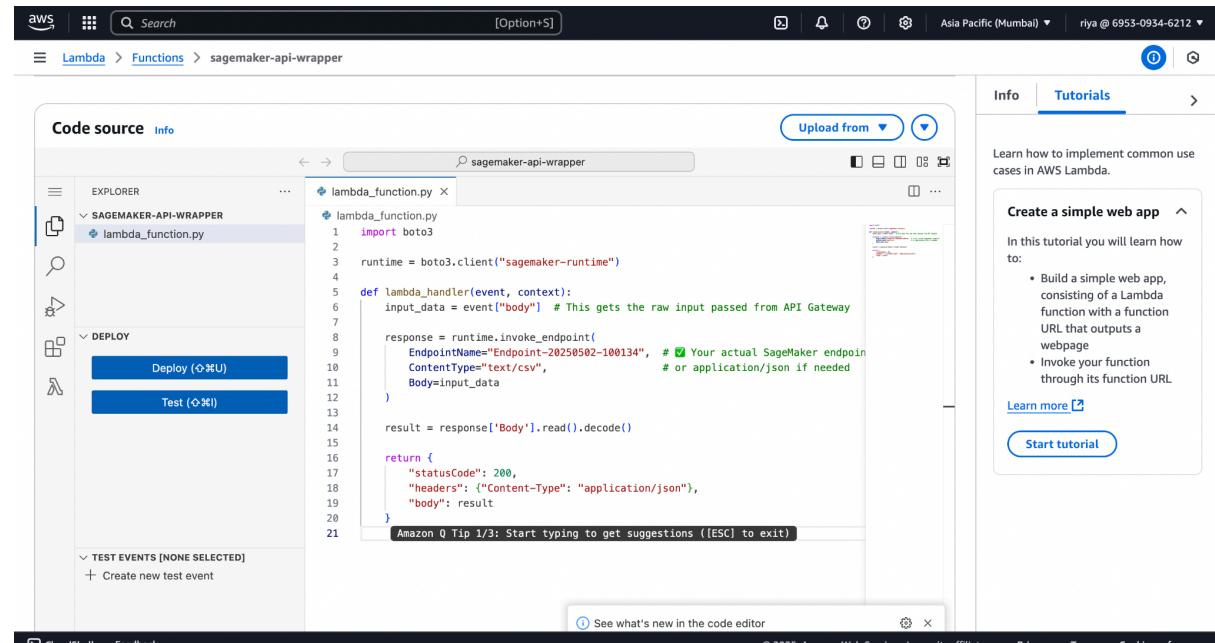
To find the inference component go to Jupyterlab in Amazon sagemaker AI and open a python3 kernel-

Then use the following code to get your inference component name-

```

paginator = client.getPaginator("list_inference_components")
for page in paginator.paginate():
    for component in page["InferenceComponents"]:
        print(component["InferenceComponentName"])

```



Then go to add trigger

Screenshot of the AWS Lambda Function Overview page for "sagemaker-api-wrapper".

**Function Overview:**

- Description:** Last modified 41 seconds ago.
- Function ARN:** arn:aws:lambda:ap-south-1:695309346212:function:sagemaker-api-wrapper
- Function URL:** -

**Code Source:** Info

**Add trigger:**

**Trigger configuration:** Info

Select a source: API Gateway

**APIs/Interactive/web:**

- Alexa
- API Gateway
- Application Load Balancer
- CodeCommit
- Cognito Sync Trigger
- VPC Lattice

**Batch/bulk data processing:**

- AWS IoT
- CloudWatch Logs
- EventBridge (CloudWatch Events)

**Tutorials:** Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

[Learn more](#) [Start tutorial](#)

Select API gateway as the source then select creating new API as intent and API Type- REST API

15- To add permissions- go to configuration

aws Search [Option+S] Lambda > Functions > sagemaker-api-wrapper

API Gateway (2) + Add destination + Add trigger

Code Test Monitor Configuration Aliases Versions

General configuration Info

Description - Memory 128 MB Ephemeral storage 512 MB

Timeout 0 min 3 sec SnapStart Info None

Edit

Learn how to implement common use cases in AWS Lambda.

Create a simple web app ^

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more Start tutorial

https://ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#/functions/sagemaker-api-wrapper?tab=configure © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Then click on permissions and click on the role name link

aws Search [Option+S] Lambda > Functions > sagemaker-api-wrapper

+ Add trigger

Function URL Info -

Code Test Monitor Configuration Aliases Versions

General configuration Info

Role name sagemaker-api-wrapper-role-h7fpk337

Resource summary

To view the resources and actions that your function has permission to access, choose a service.

AWS Application Auto Scaling 10 actions, 1 resource

By action By resource

Resource	Actions
All resources	Allow: application-autoscaling:PutScalingPolicy Allow: application-autoscaling:DescribeScalingActivities Allow: application-autoscaling:DescribeScalingPolicies Allow: application-autoscaling>DeleteScheduledAction Allow: application-autoscaling:PutScheduledAction Allow: application-autoscaling>DeleteScalingPolicy

View role document

Learn how to implement common use cases in AWS Lambda.

Create a simple web app ^

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more Start tutorial

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Then click on add permissions- attach policies and search for AmazonSageMakerFullAccess and add it

Screenshot of the AWS IAM Roles page showing the details of the 'sagemaker-api-wrapper-role-h7fpk337' role.

**Identity and Access Management (IAM)**

**sagemaker-api-wrapper-role-h7fpk337**

**Summary**

**Creation date:** May 02, 2025, 17:50 (UTC+05:30)

**Last activity:** 17 hours ago

**ARN:** arn:aws:iam::695309346212:role/service-role/sagemaker-api-wrapper-role-h7fpk337

**Maximum session duration:** 1 hour

**Permissions**

**Permissions policies (3)**

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AllowSageMakerInvoke	Customer inline	0
AmazonSageMakerFullAccess	AWS managed	12
AWSLambdaBasicExecutionRole-c988a...	Customer managed	1

<https://us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/details/sagemaker-api-wrapper-role-h7fpk337/attach-policies>

## 16- Go to API gateway.

Screenshot of the AWS API Gateway landing page.

**Networking & Content Delivery**

**API Gateway**

Create and manage APIs at scale

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs.

**How it works**

API Gateway enables you to connect and access data, business logic, and functionality from backend services such as workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, any web application, or real-time communication applications.

**Get started**

Create a new API to begin exploring API Gateway. You can also import an external definition file into API Gateway.

[Create an API](#)

**Pricing**

With Amazon API Gateway, you only pay when your APIs are in use. There are no minimum fees or upfront commitments. For HTTP and REST APIs, you pay based on API calls received and amount of data transferred out. For WebSocket APIs, you pay based on number of messages and connection duration.

[Learn more about pricing](#)

**Resources**

- [Getting Started Guide](#)
- [Developer Guide](#)
- [API References](#)

© 2025, Amazon Web Services, Inc. or its affiliates.

## Click on Create API

The screenshot shows the AWS API Gateway interface. On the left, a sidebar titled 'API Gateway' has a 'APIs' section with links to 'Custom domain names', 'Domain name access associations', 'VPC links', 'Usage plans', 'API keys', 'Client certificates', and 'Settings'. The main content area is titled 'APIs (1/1)' and shows a table with one row. The table columns are 'Name', 'Description', 'ID', 'Protocol', and 'API endpoint type'. The row contains the values: 'sagemaker-api-wrapper-API', 'Created by AWS Lambda', 'rqaiairffj', 'REST', and 'Regional'. At the top right of the table are 'Delete' and 'Create API' buttons. Below the table is a search bar with placeholder text 'Find APIs'.

## Choose REST API and then build

The screenshot shows the 'Create API' wizard. The first step, 'WebSocket API', is selected. It contains a brief description: 'Build a WebSocket API using persistent connections for real-time use cases such as chat applications or dashboards.' Below this is a note: 'Works with the following: Lambda, HTTP, AWS Services' and a 'Build' button. The second step, 'REST API', is shown below it. Its description is: 'Develop a REST API where you gain complete control over the request and response along with API management capabilities.' It also includes the note: 'Works with the following: Lambda, HTTP, AWS Services' and 'Import' and 'Build' buttons. The third step, 'REST API Private', is at the bottom. Its description is: 'Create a REST API that is only accessible from within a VPC.' It includes the note: 'Works with the following: Lambda, HTTP, AWS Services' and 'Import' and 'Build' buttons. At the very bottom of the screen, there is a footer bar with links for 'CloudShell', 'Feedback', '© 2025, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

Screenshot of the AWS API Gateway 'Create REST API' wizard and the resulting API resource page.

**Create REST API**

**API details**

- New API Create a new REST API.
- Clone existing API Create a copy of an API in this AWS account.
- Import API Import an API from an OpenAPI definition.
- Example API Learn about API Gateway with an example API.

**API name**: My\_demo\_api

**Description - optional**: (Empty)

**API endpoint type**: Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.

**Regional**

**IP address type**:  
 **IPv4** Supports only edge-optimized and Regional API endpoint types.  
 **Dualstack** Supports all API endpoint types.

**CloudShell** **Feedback** © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

**Search** [Option+S] AWS Asia Pacific (Mumbai) rya @ 6953-0934-6212

**API Gateway** > **APIs** > Resources - DEMO2\_API (t1dx5cguzl)

**API Gateway**

- APIs
- Custom domain names
- Domain name access associations
- VPC links

**▼ API: DEMO2\_API**

- Resources** (Selected)
- Stages
- Authorizers
- Gateway responses
- Models
- Resource policy
- Documentation
- Dashboard
- API settings

**Usage plans**

**Resources**

Successfully created REST API 'DEMO2\_API (t1dx5cguzl)'.

**Resource details**

- Path: /
- Resource ID: g9mcwn9vni
- Update documentation
- Enable CORS

**Methods (0)**

Method type	Integration type	Authorization	API key
No methods defined.			

**API actions** Deploy API

**CloudShell** **Feedback** © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Choose POST method and then integration type as Lambda function and then connect the lambda function that we created in the link below

**Create method**

**Method details**

Method type: POST

Integration type:

- Lambda function: Integrate your API with a Lambda function. (Selected)
- HTTP: Integrate with an existing HTTP endpoint.
- Mock: Generate a response based on API Gateway mappings and transformations.
- AWS service: Integrate with an AWS Service.
- VPC link: Integrate with a resource that isn't accessible over the public internet.

**Lambda proxy integration**: Send the request to your Lambda function as a structured event.

**Lambda function**: Provide the Lambda function name or alias. You can also provide an ARN from another account.

arn:aws:lambda:ap-south-1:695309346212:function:sagemaker-api-wrapper

Click on deploy API

**API Gateway**

APIs  
Custom domain names  
Domain name access associations  
VPC links

**API: sagemaker-api**

Resources  
Stages  
Authorizers  
Gateway responses  
Models  
Resource policy  
Documentation  
Dashboard  
API settings

Usage plans  
API keys  
Client certificates  
Settings

**Resources**

Successfully created method 'POST' in '/'. Redeploy your API for the update to take effect.

**/ - POST - Method execution**

ARN: arn:aws:execute-api:ap-south-1:695309346212:cm2zxfzca7/\*/POST/

Resource ID: 2nwd44w6k1

Method request → Integration request → Integration response → Method response

Client ← Method response ← Integration response ← Integration request ← Method request

**Method request settings**

Authorization: NONE

Request validator: None

API key required: False

SDK operation name: None

Copy the URL after deploying the API to use on postman

The screenshot shows the AWS API Gateway interface. On the left, a sidebar lists 'APIs', 'Custom domain names', 'Domain name access associations', 'VPC links', and an expanded 'API: sagemaker-api' section containing 'Resources', 'Stages' (which is selected), 'Authorizers', 'Gateway responses', 'Models', 'Resource policy', 'Documentation', 'Dashboard', and 'API settings'. Below this are 'Usage plans', 'API keys', 'Client certificates', and 'Settings'. At the bottom of the sidebar are 'CloudShell' and 'Feedback' buttons. The main content area has a header 'Stages' with a 'Stage actions' dropdown and a 'Create stage' button. A green banner at the top says 'Successfully created deployment for sagemaker-api. This deployment is active for sagemaker\_demo\_API.' Below the banner is a 'Notifications' section with 0 notifications. The 'sagemaker\_demo\_API' stage is listed with 'Stage details': Stage name 'sagemaker\_demo\_API', Rate 'Info 10000', Cache cluster 'Info Inactive', Default method-level caching 'Info Inactive', and a 'Copied' message above a 'Take URL' button. To the right are 'Web ACL' and 'Client certificate' sections, both marked as '-'. Below the stage details is an 'Active deployment' section showing 'gxdqds on May 02, 2025, 18:00 (UTC+05:30)'. At the bottom of the main content area are 'Logs and tracing' (CloudWatch logs 'Info Inactive', Detailed metrics 'Info Inactive', Data tracing 'Info Inactive'), a copyright notice '© 2025, Amazon Web Services, Inc. or its affiliates.', and links for 'Privacy', 'Terms', and 'Cookie preferences'.

## 17- Test in Postman-

Choose POST method and then enter your URL

Then enter the following in the body and click send to get your prediction-

```
{  
  "body": {  
    "ID": 1,  
    "Age": 24,  
    "Experience": 4,  
    "Income": 300000,  
    "Zipcode": 400053,  
    "Family": 1,  
    "CCAvg": 400,  
    "Education": 4,  
    "Mortgage": 0,  
    "SecuritiesAccount": 1,  
    "CDAccount": 1,  
    "Online": 1,  
    "CreditCard": 0  
  }  
}
```

The screenshot shows the Postman application interface. On the left, the sidebar includes 'My Workspace' (Collections, Environments, Flows, History), 'Create a collection for your requests', and a note about collections. The main area displays a collection named 'My first collection' containing two folders: 'First folder inside collection' and 'Second folder inside collection'. A specific request is selected, showing its details: Method: POST, URL: https://cm2zxfzca7.execute-api.ap-south-1.amazonaws.com/Postman\_API. The 'Body' tab is active, showing raw JSON data:

```
1 {
2   "ID": 1,
3   "Age": 24,
4   "Experience": 4,
5   "Income": 300000,
6   "Zipcode": 400053,
7   "Family": 1,
8   "CCAvg": 400,
9   "Education": 4,
10  "Mortgage": 0,
11  "SecuritiesAccount": 1,
12  "CDAccount": 1,
```

The response status is 200 OK, with a duration of 1.30 s and a size of 508 B. The response body is also shown in JSON format:

```
1 {
2   "predictions": [
3     {
4       "predicted_label": "1.0",
5       "probability": 0.9950631856918335,
6       "probabilities": "[0.9950631856918335, 0.004936841782182455]",
7       "labels": "[1.0, 0.0]"
8     }
9   ]
10 }
```