**Tabular Prediction with SageMaker AutoPilot Reference App**

Tabular AI prediction refers to the application of artificial intelligence techniques to structured, table-like data for making predictions or forecasts. Enterprise today has massive historic data, and the use of AI in this context, especially machine learning prediction models, can provide valuable insights and predictions from the data. Here are some common use cases included in the reference app:

* 1. Financial Services: In a role of a business analyst working for a company in the banking sector. Your goal is to predict whether your customer will repay the loan taken, or not.
* 2. Insurance: Multiple linear regression based on several features of individual against their existing medical expense to be used for predicting future medical expenses of individuals that help medical insurance to make decision on charging the premium.
* 3. Manufacturing: As a business analyst of a large manufacturing org, to predict which machine failure will occur in the future given a historical dataset that contains characteristics tied to a given failure type.
* 4. Marketing: As a business analyst of a large manufacturing org, to predict which machine failure will occur in the future given a historical dataset that contains characteristics tied to a given failure type.
* 5. Healthcare & Life Sciences: To predict the likelihood of diabetes in patients based on their medical history and demographic details.
* 6. Service: Predict the vehicle maintenance service given their historical service data

These are just a few examples. The key advantage of AI in tabular data prediction is its ability to uncover complex patterns and relationships within the data that might not be apparent through traditional analysis methods.

**Features & Functionality**

Appian tabular prediction with SageMaker reference app uses Amazon SageMaker Autopilot to automatically build, train, and tune the best custom machine learning (ML) models based on your data. It’s an automated machine learning (AutoML) solution that eliminates the heavy lifting of handwritten ML models that requires ML expertise. Users need to only provide a tabular dataset(csv format) and select the target column to predict, and Autopilot automatically infers the problem type, performs data preprocessing and feature engineering, selects the algorithms and training mode, and explores different configurations to find the best ML model. The problem type can be either binary classification, multiclass classification, or regression.

Key features:

* - Streamline SageMaker model creation, train, tune, and deployment
* - Support endpoint and model deletion - List all models with endpoints in one screen
* - Administrative tasks such as remove model and purge endpoint of xx days old
* - 6 use cases across 6 difference industries are included, along with their datasets, original dataset link with description, and model report to show model quality
* - For the first 5 use cases except the service use case, datasets are provided in the download zip file

How it works

* AWS SageMaker AutoPilot Tech Stack

GitHub: <https://github.com/appian/sagemaker-autopilot-sample-solution>

A diagram of a software structure

Description automatically generated

**Pre-requisite**

* An S3 Bucket ready to upload SageMaker stack code. Details see Deployment Stack Step 1 below.
* AWS CLI has been installed locally.

To download and install AWS CLI, click [link](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html) here.

* Your AWS account with either Admin role or role with permission for SageMaker, S3, SNS, Lambda
* AWS Account region, Access Key Id, and Secret Access Key
* AWS S3 Connected System Plugin
* AWS Lambda Connected System Plugin

**Deployment for AWS AutoPilot Stack**

1. Download the above Github code to a local drive e.g. C:\sagemaker
2. Create a S3 bucket “xxs-sgmkr-artifacts” to store the AWS code (for a big group of users, strongly suggest use an initial such as “<FirstName><1st character of LastName>”, mine will be: philipk)
3. Create a new Standard SNS Topic “xx-automl-topic” and add an Email subscription to it. Take notes of the Topic ARN and the Subscription ID.

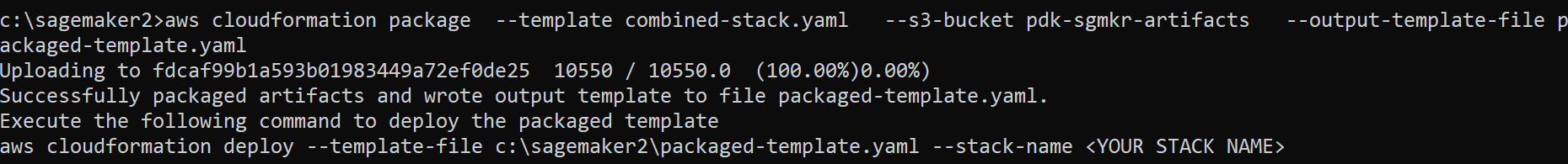
A screenshot of a computer

Description automatically generated

1. Run “aws configure” with your Access Key and Secret Access Key, make sure the region is set correctly where the SageMaker service is available.
2. Package and upload AWS code artifacts to the S3 bucket (created in step 2)

In windows:

C:\sagemaker> aws cloudformation package --template combined-stack.yaml --s3-bucket xx-sgmkr-artifacts --output-template-file packaged-template.yaml



1. Deploy Cloudformation template

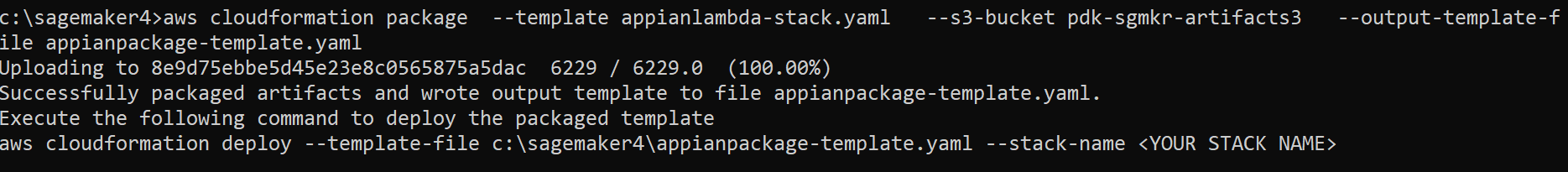
aws cloudformation deploy --template packaged-template.yaml --stack-name *sagemaker-automl-sample* --parameter-overrides EnvName=*xxx-automl* LambdaFunctionName=*xxx-automl-pipeline-fn* SourceCodeBucket=*xxx-automl-sourcecode-bucket* DataBucket=*xxx-automl-data-bucket* ModelBucket=*xxx-automl-model-bucket* TargetID=<*SNS Subscription ID*> SNSARN=<*SNS Topic ARN*> PipelineName=*xxx-automlpipeline* --capabilities CAPABILITY\_NAMED\_IAM

A computer screen with white text

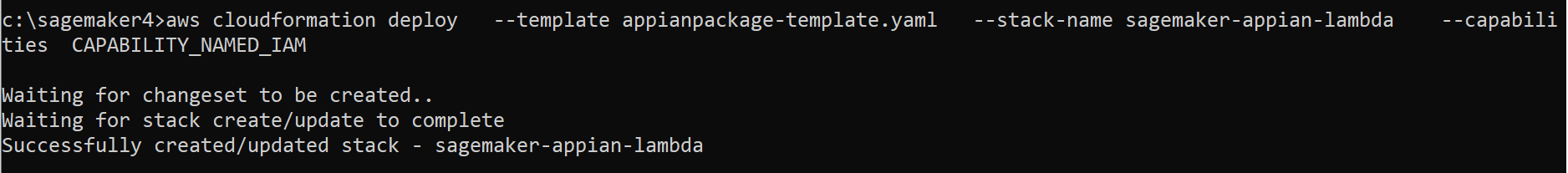
Description automatically generated

1. Deploy Appian’s Lambda Functions

aws cloudformation package --template appianlambda-stack.yaml --s3-bucket pdk-sgmkr-artifacts3 --output-template-file appianpackage-template.yaml



aws cloudformation deploy --template appianpackage-template.yaml --stack-name sagemaker-appian-lambda --capabilities CAPABILITY\_NAMED\_IAM



1. Config Appian App to reflect the S3 bucket
   1. Modify Integration Object “SM\_StartModelTraining”, change Function Name to the ones of your AWS account.

* SM\_StartModelTraining -- change Function Name to : invokesmpipeline-automl-<xxx-automl>
* SM\_DeployEndpoint – change Function Name to: deploysmept-automl-<xxx-automl>
  1. After Step 6 execution, it will create 3 S3 buckets. One of the bucket will be the “xxx-automl-data-bucket”, which you will need to note down and type in during the model training step later.

1. Modify Connected Systems setting
   1. “SM Lambda Sandbox CS” to use your own AWS Account credentials for lambda functions integration
   2. “SM\_S3” to use your own AWS Account credentials for S3 integration
   3. “SM Sandbox SageMaker CS” to use your own AWS Account credentials for Model Inference prediction
2. Run SQL scripts to create 6 DB tables. (sql scripts are included in the zip file)

**How to use the Reference App**

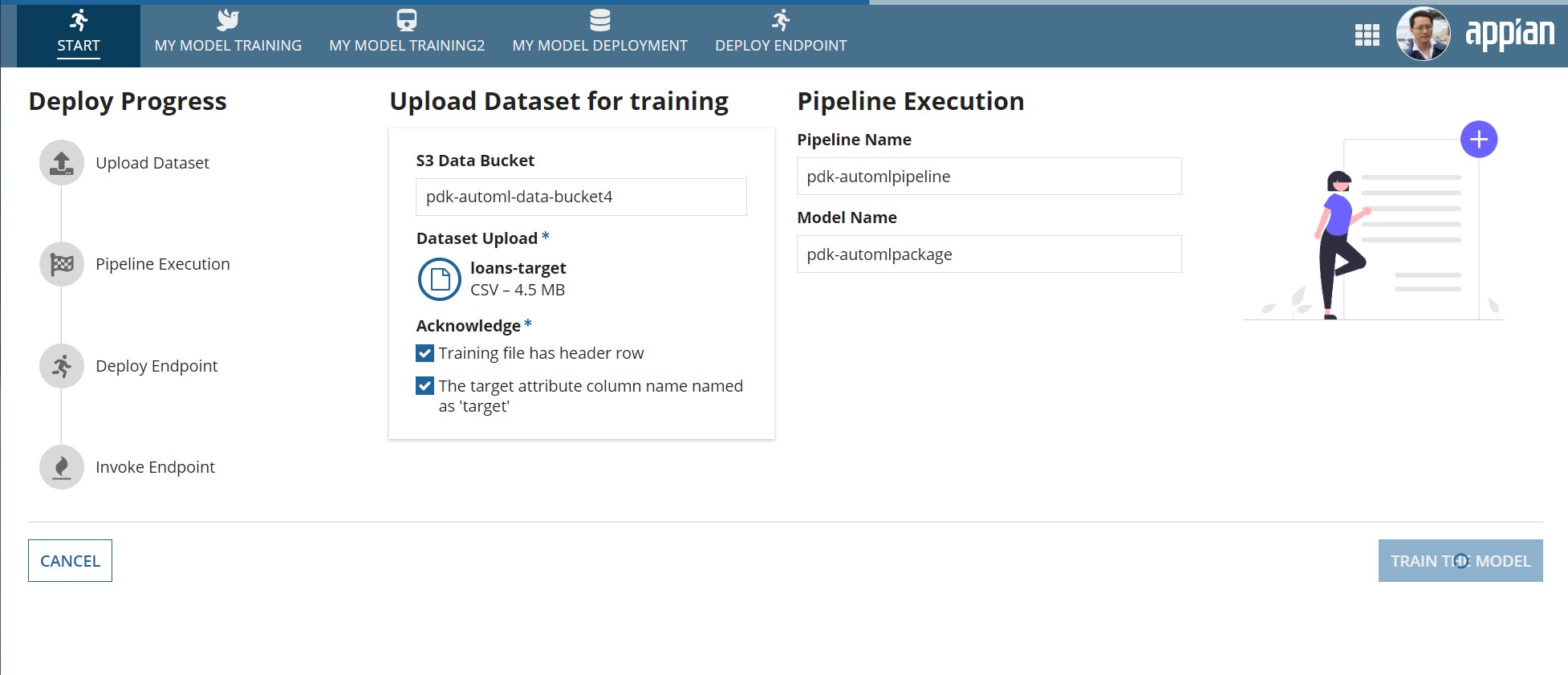
* CSV Dataset requirements
  + Has a header row
  + Target column header need to rename as “target”
* Mandatory to use prefix for model name and endpoint name
  + For a big team of users, please use a prefix to clearly identify whose model it belongs to.
  + For example, <firstName><1st Character of LastName> as prefix, e.g. “philipk”

**Reference App Demo Video**

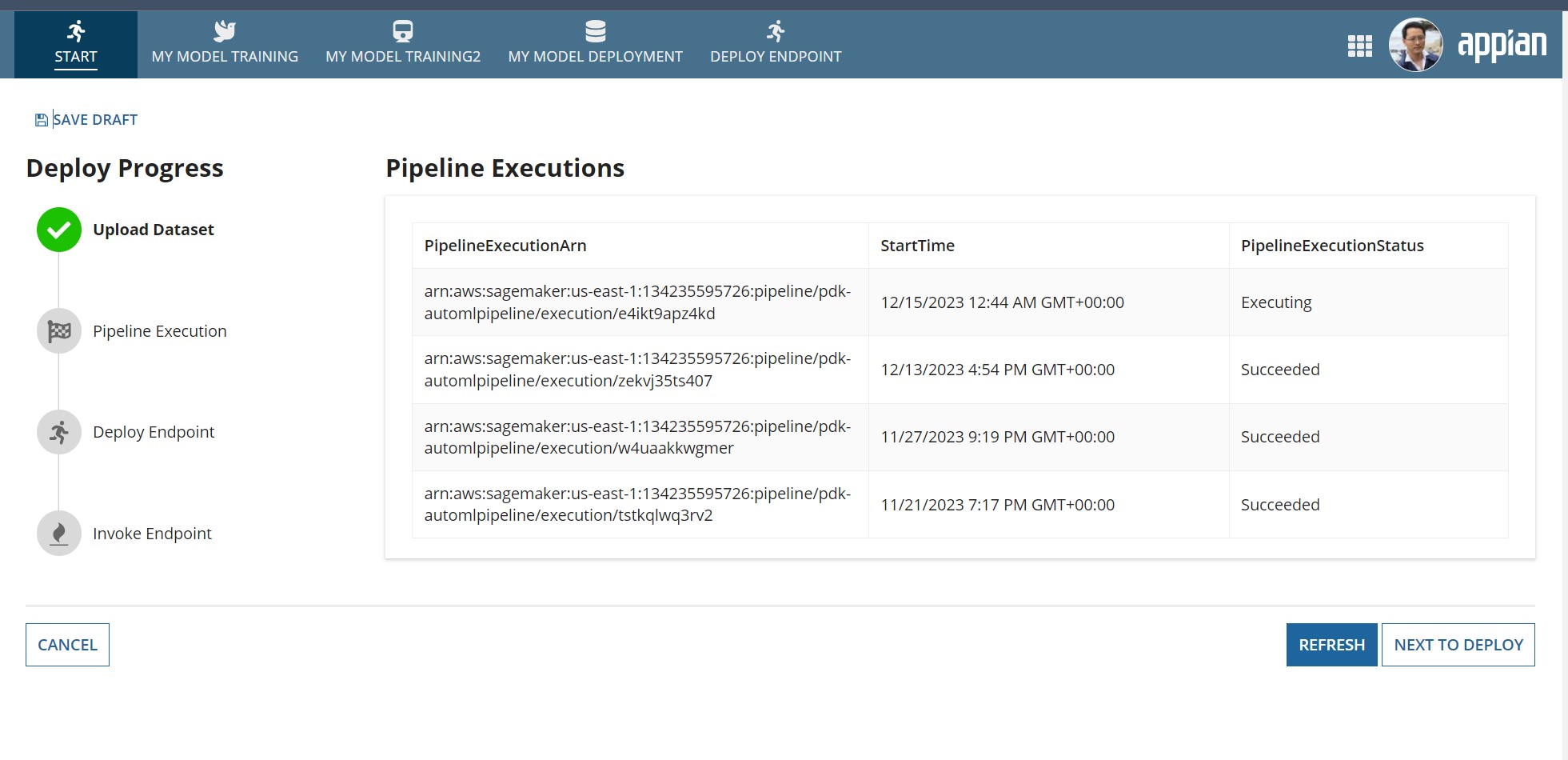
<https://youtu.be/HgIXrMU6-Xs>

**Create model and deploy endpoint**

* Navigate to Start Tab
* Upload the data file in csv format
  + Make sure the csv file has the to-predict column header named as “target”



Pipeline Executions (model training status)

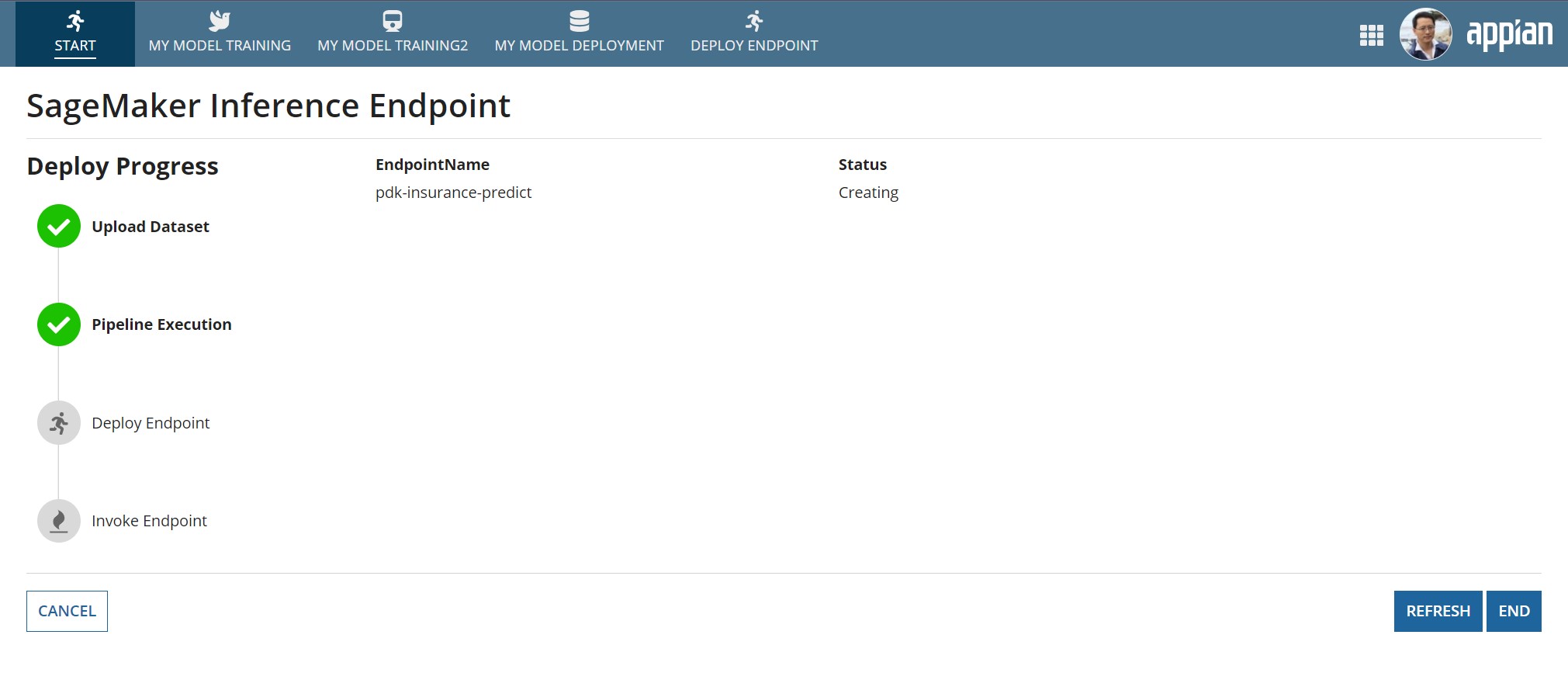


Deploy Endpoint

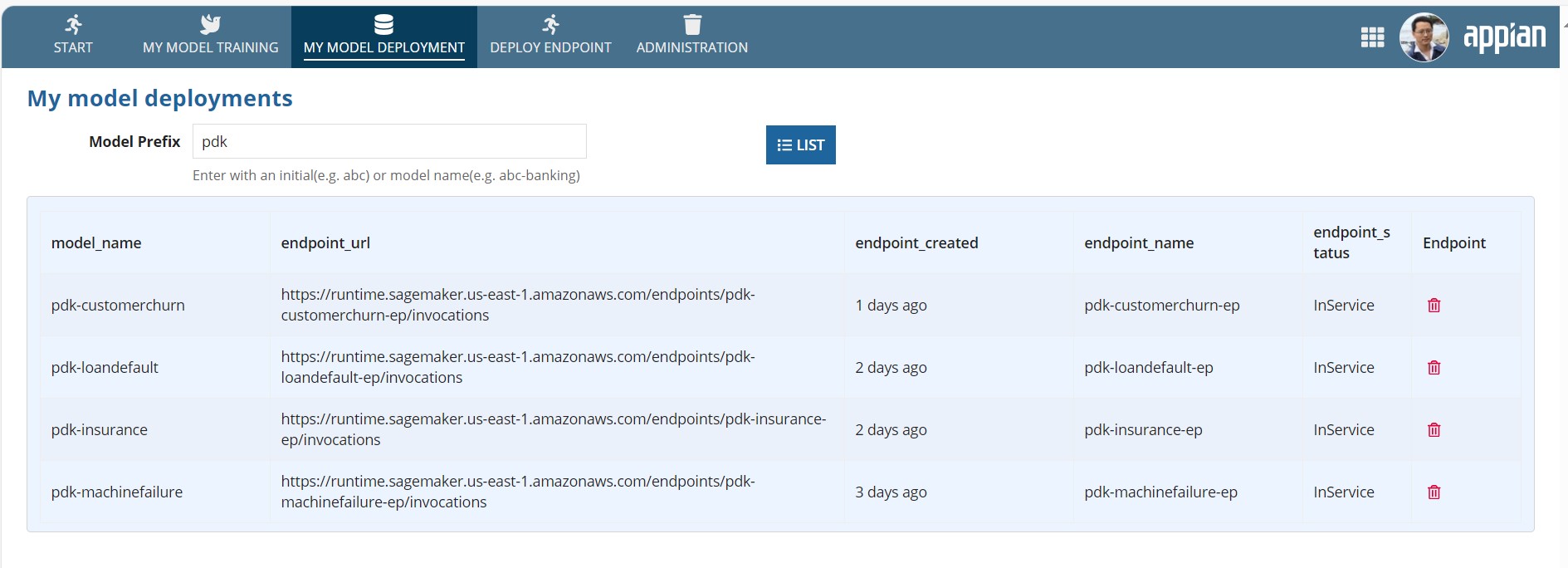
A screenshot of a computer

Description automatically generated

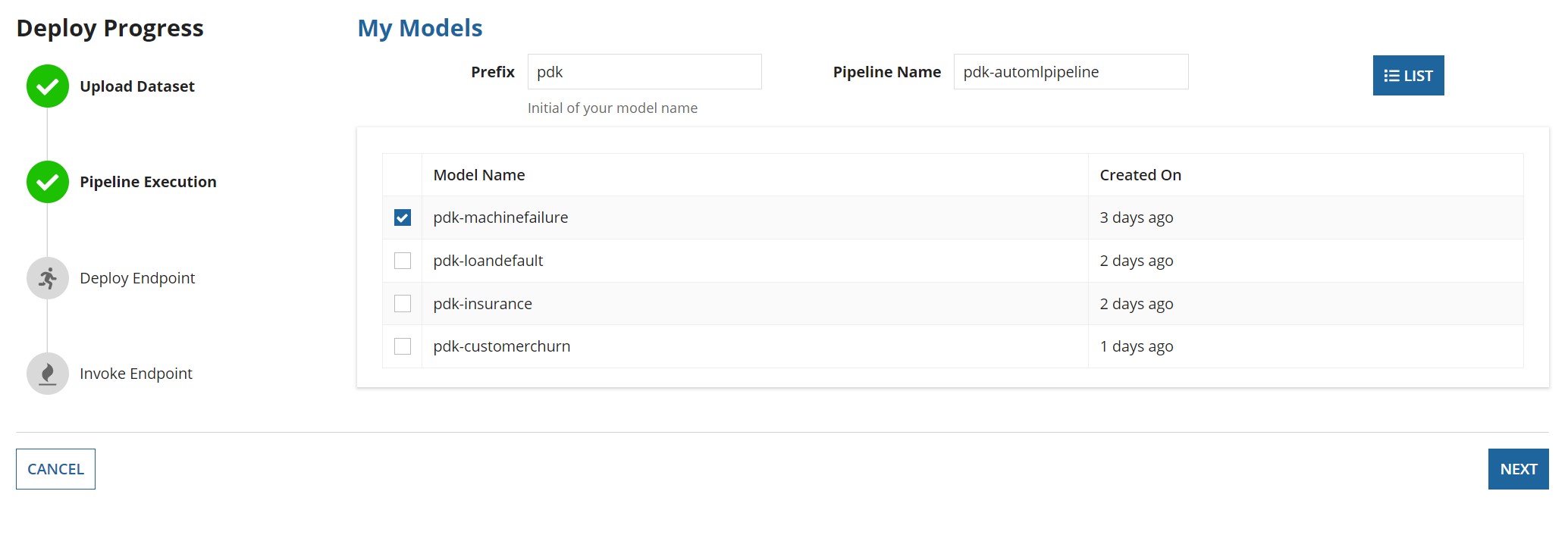
* Deploy Status

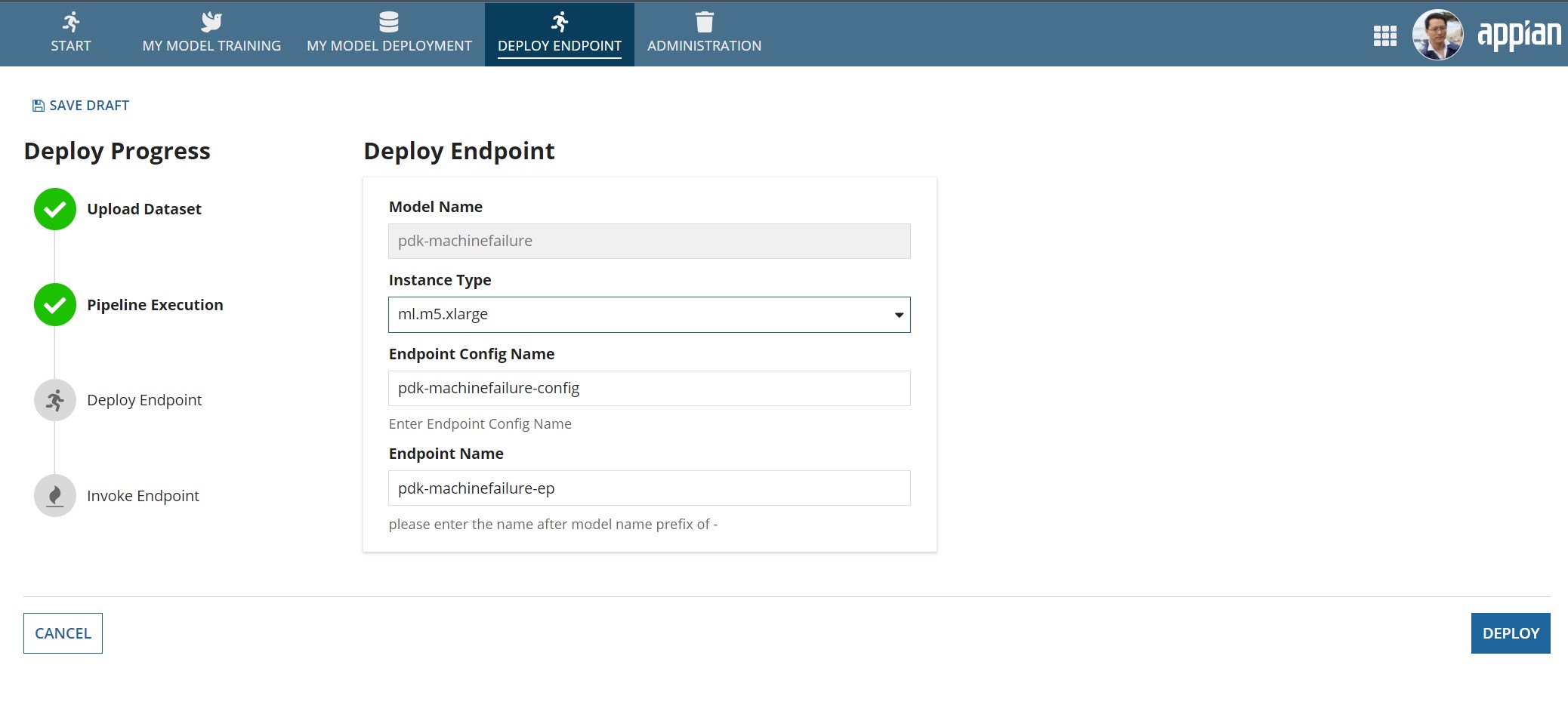


* Check My Model Deployment



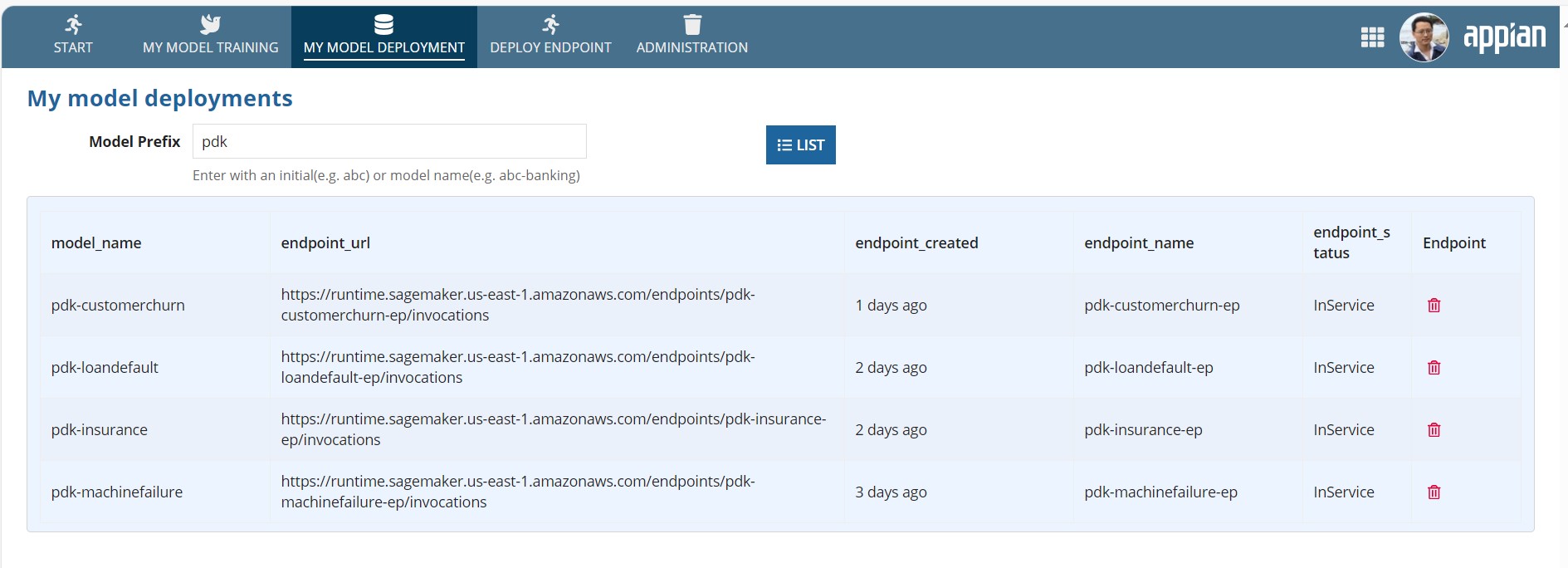
**Deploy Endpoint separately**

****

****

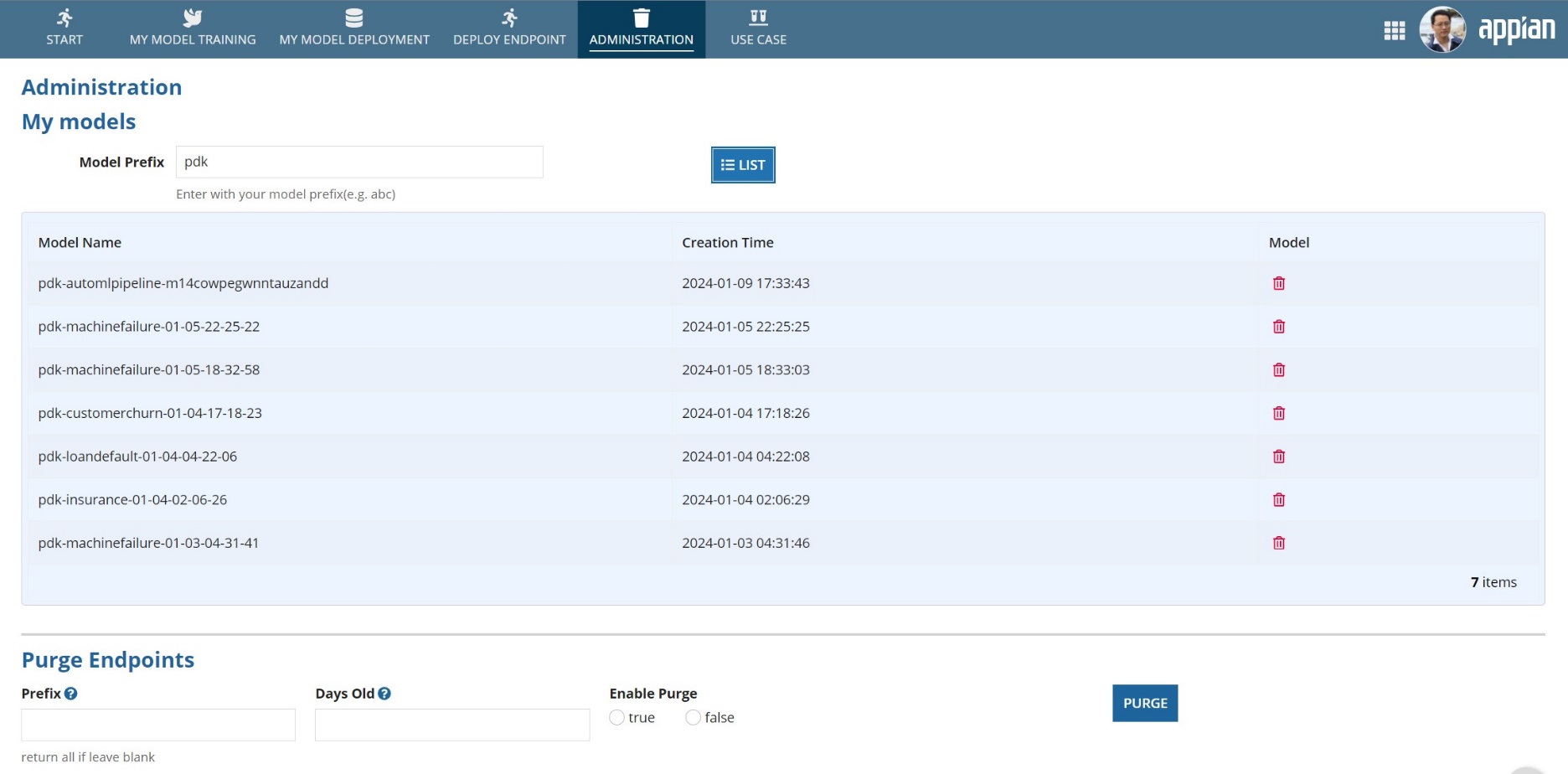
Video: <https://youtu.be/lFxpIKRqaJI>

**My Model Deployment**

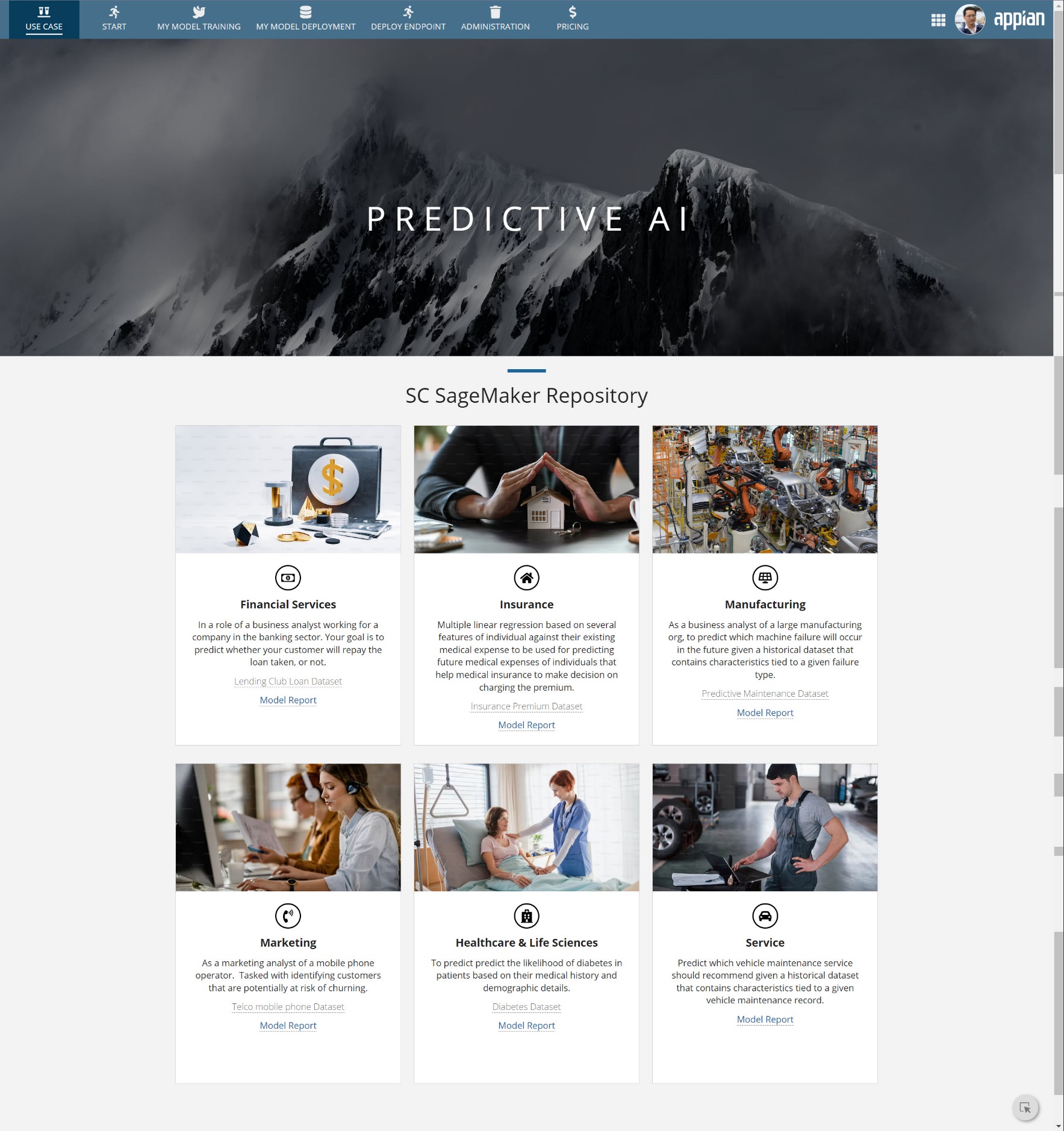


**Administration**

Administration is meant to use by Administrators. Administrators can delete models and purge endpoints by xxx days old.

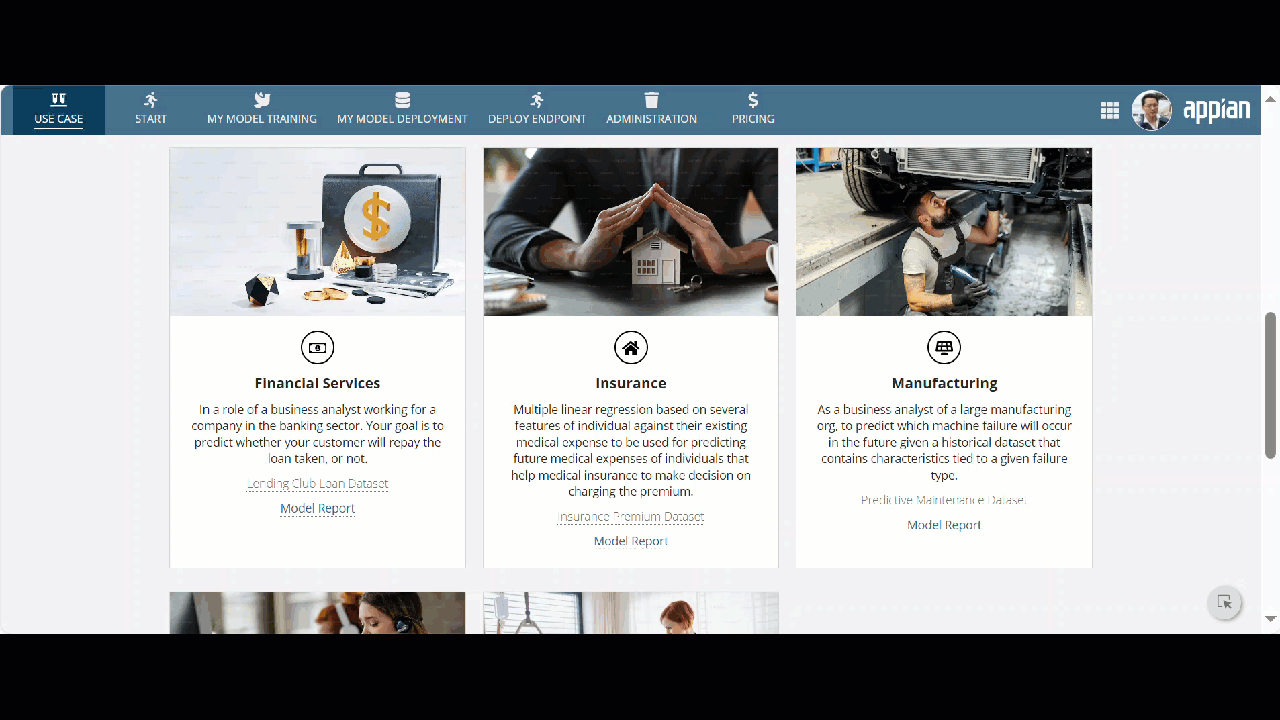
****

**Use Case**



Use Case Video

Video: <https://youtu.be/mfyZOnPmzyQ>

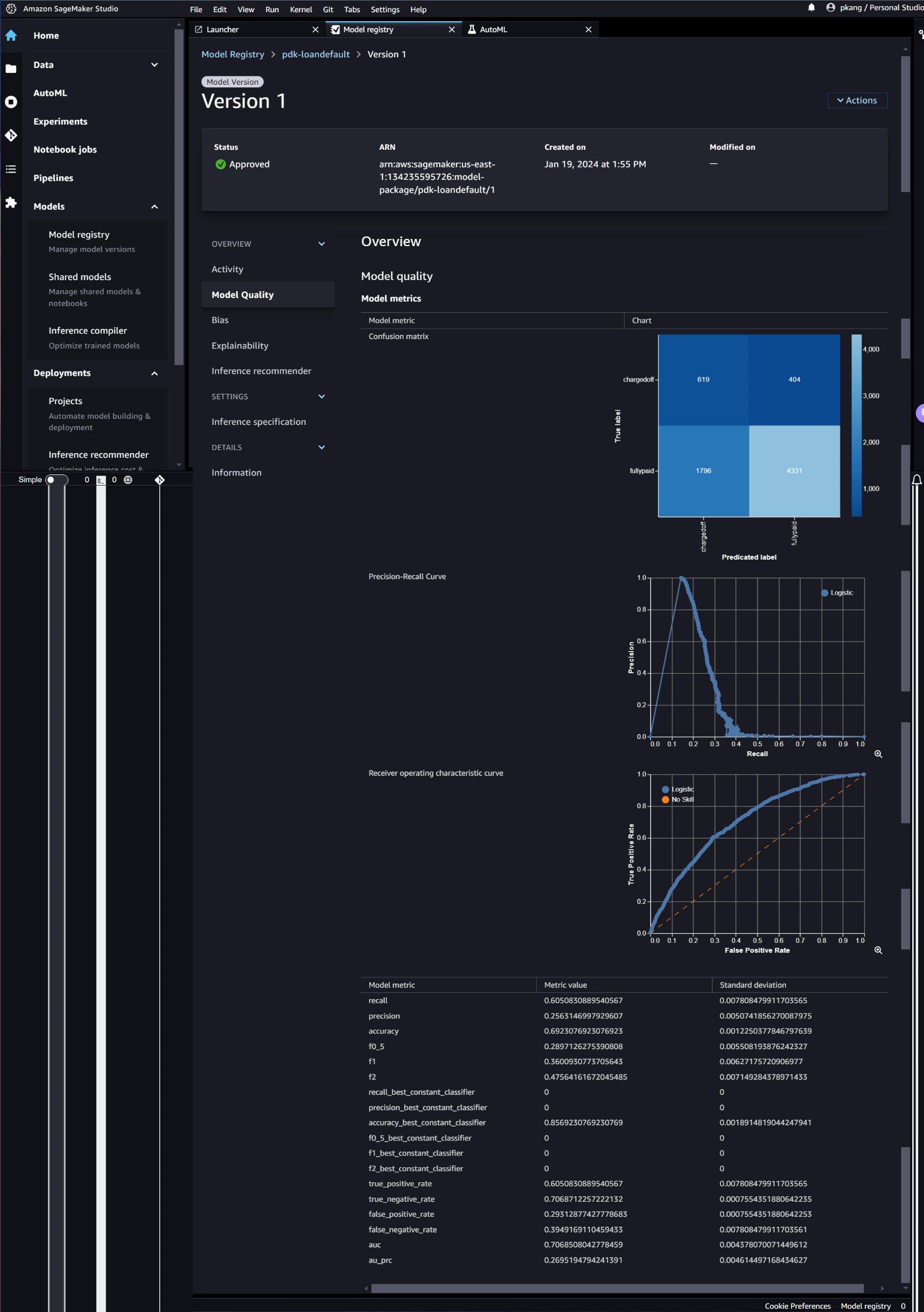


Configuration for use case tab

* Train 6 models using dataset provided in the reference app download.
* After models have been trained and deployed, change following integrations accordingly
  + Financial Services: SM\_sandbox\_loandefault\_endpoint\_new
  + Insurance: SM\_sandbox\_insurance\_predict
  + Manufacturing: SM\_sandbox\_machinefailure\_predict
  + Marketing: SM\_sandbox\_customerchurn\_predict
  + Healthcare & Life Sciences: SM\_sandbox\_diabetes\_predict
  + Service: SM\_sandbox\_vehicle\_predict
* You can either obtain the model report as described below from your own SageMaker Studio, or you can use the supplied model reports.
  + To show the supplied model reports
    1. Modify Interface object “SM\_UseCase\_Repository” to change the uri link to your own cloud instance

e.g. change from “https://xxx.appiancloud.com” to “https://<your apian cloud instance>”

**How to obtain the SageMaker Model Quality metrics and report**

* Login to your own AWS account
* Navigate to SageMaker >> Amazon SageMaker Studio >> Models >> latest version of selected model >> Model Quality
* 
* Navigate down to “Explainability” to download dataset feature report in PDF format.

A screenshot of a computer

Description automatically generated

**Model Metrics**

To understand Autopilot metrics and Autopilot weighted metrics, please check out the Amazon SageMaker documentation

Link: <https://docs.aws.amazon.com/sagemaker/latest/dg/autopilot-metrics-validation.html>