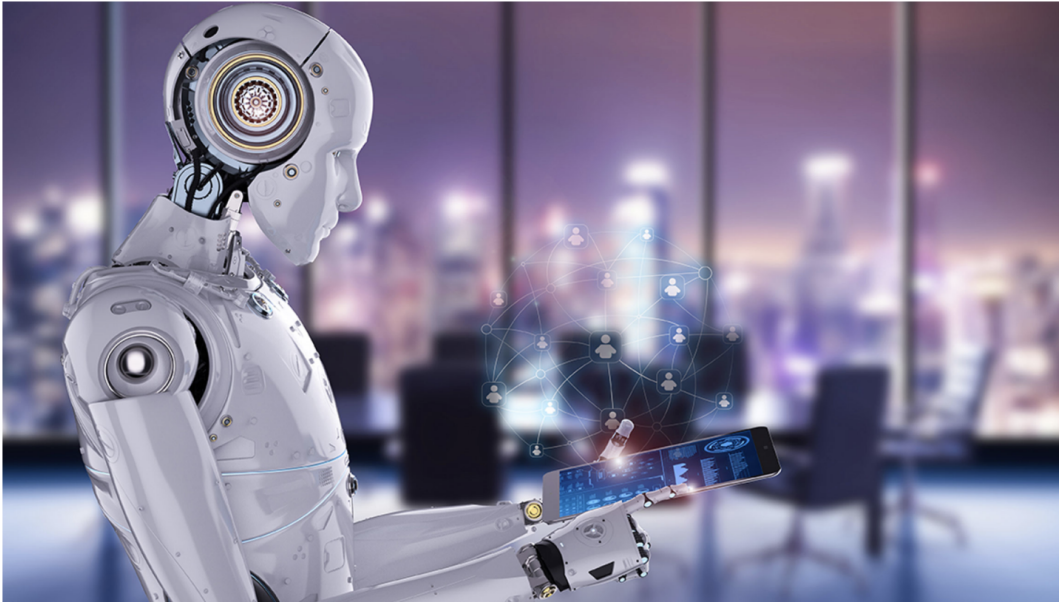


# Robo Advisor

□ Challenge 15



"In this Challenge, I combine my new AWS (Amazon Web Services) skills with existing Python superpowers to create a bot that will recommend an investment portfolio for a retirement plan "

## Table of content

- [Overview of the project and project goals](#)
- [Software version control](#)
  - [Libraries](#)
  - [Work with GitHub](#)
  - [How to install](#)
- [Working Bot Demonstration](#)
- [Helps recruiters](#)
- [License](#)

## Overview of the project and project goals

Using the power of machine learning and NLP to disrupt finance and improve the customer experience, we are creating a robo advisor. Both existing and potentially new customers will be able to use this robo advisor to get investment portfolio recommendations for retirement.

Following main tasks were accomplished:

1. Configure the initial robo advisor: Define an Amazon Lex bot with a single intent that establishes a conversation about requirements to suggest an investment portfolio for retirement.
2. Build and test the robo advisor: Make sure that your bot works and accurately responds during the conversation with the user.
3. Enhance the robo advisor with an Amazon Lambda function: Create an Amazon Lambda function that validates the user's input and returns the investment portfolio recommendation. This includes testing the Amazon Lambda function and integrating it with the bot.

## Software version control


Amazon Lex is used to create the bot

### Step 1: Initiate the bot

**Bot name**


**Language**


**Output voice**




**Session timeout**   

**Sentiment analysis** ☐ Yes ☒ No 

**IAM role** [AWSServiceRoleForLexBots](#)   
Automatically created on your behalf

**COPPA** Please indicate if your use of this bot is subject to the [Children's Online Privacy Protection Act](#) (COPPA). [Learn more](#)   
☐ Yes ☒ No

**Advanced options** Enable accuracy improvements and ML features. [Learn more](#)   
☐ Yes ☒ No

**Confidence score threshold**

## Step 2: Create the intent

recommendPortfolio Latest ▼

### ▼ Sample utterances ⓘ

+

✕

✕

✕

✕

✕

✕

✕

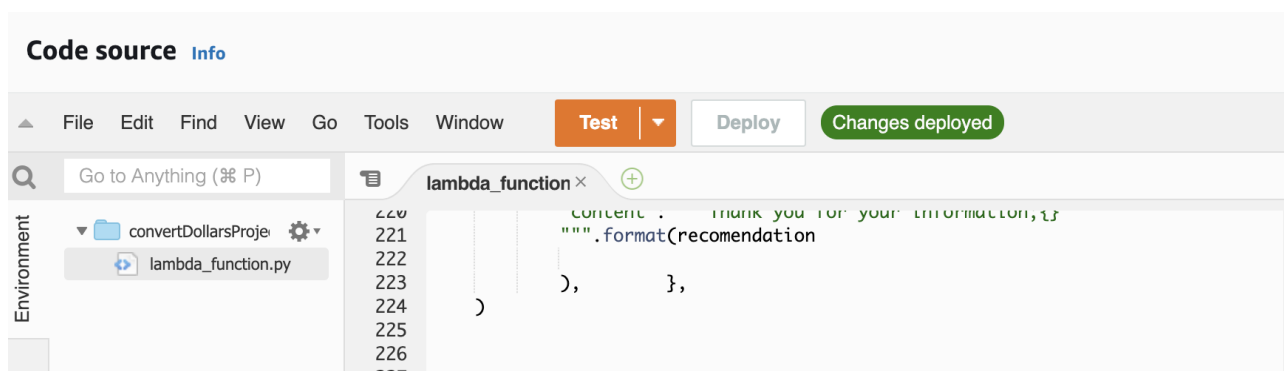
### ▶ Lambda initialization and validation ⓘ

### ▼ Slots ⓘ

Priority	Required	Name	Slot type	Version	Prompt
		<div>e.g. Location</div>	<div>e.g. AMAZON.U... ▾</div>		<div>e.g. What city?</div>
1.	▾	<div><input checked="" type="checkbox"/></div> <div>firstName</div>	<div>AMAZON.DE_FI... ▾</div>	<div>Built-in ▾</div>	<div>Thank you for trusting me to hel</div>
2.	^ ▾	<div><input checked="" type="checkbox"/></div> <div>age</div>	<div>AMAZON.NUM... ▾</div>	<div>Built-in ▾</div>	<div>How old are you?</div>
3.	^ ▾	<div><input checked="" type="checkbox"/></div> <div>investmentAmount</div>	<div>AMAZON.NUM... ▾</div>	<div>Built-in ▾</div>	<div>How much do you want to inves</div>
4.	^	<div><input checked="" type="checkbox"/></div> <div>riskLevel</div>	<div>AMAZON.Alpha... ▾</div>	<div>Built-in ▾</div>	<div>What level of investment risk wc</div>

## Step 3: Create Lambda

Amazon Lambda is used to enhance original bot



## Step 4: Test Lambda

You can find Test Files in the repo in separate folder. On the following picture "negative age" mistake was tested and code passed.

**Code source** [Info](#)

File Edit Find View Go Tools Window **Test** Deploy Changes deployed

Go to Anything (% P)

Environment

- convertDollarsProje
  - lambda\_function.py

Execution results

**Test Event Name**  
negativeage

**Response**

```
{
  "sessionAttributes": {},
  "dialogAction": {
    "type": "ElicitSlot",
    "intentName": "recommendPortfolio",
    "slots": {
      "firstName": "John",
      "age": null,
      "riskLevel": "Low",
      "investmentAmount": "5000"
    },
    "slotToElicit": "age",
    "message": {
      "contentType": "PlainText",
      "content": "The person should be older than 0 and yanger 65 "
    }
  }
}
```

**Function Logs**

```
START RequestId: 26a1b435-fa35-46b6-b750-e4dd31922389 Version: $LATEST
END RequestId: 26a1b435-fa35-46b6-b750-e4dd31922389
REPORT RequestId: 26a1b435-fa35-46b6-b750-e4dd31922389 Duration: 9.42 ms Bill
```

**Request ID**  
26a1b435-fa35-46b6-b750-e4dd31922389

On the following video you can see this code deployed with the bot:

<https://user-images.githubusercontent.com/80833988/124698952-beaadd80-de9e-11eb-96a0-6cefdd348d8b.mov>

## Step 4: Deploy Lambda

### ▼ Lambda initialization and validation ⓘ

☒ Initialization and validation code hook

**Lambda function** convertDollarsProject ▼

[View in Lambda console](#) [↗](#)

**Version or alias** Latest ▼

## Libraries

- Following libraries were imported

```
# Import the required libraries and dependencies

from datetime import datetime
from dateutil.relativedelta import relativedelta
```

## Work with GitHub

- Repository created on GitHub
- Files were committed using command line
- File with code for Lambda function included lambda\_function.py

## How to install

- Save remote repo from GitHub to your computer (Desktop): in Terminal type:

```
cd desktop
```

```
git clone https://github.com/nataliaburrey/Robo_Advisor.git
```

now you can find repo on your desktop

- Choose [ lambda\_function.py ] file to see the code for Enhanced Robo Advisor.

## Working Bot Demonstration

### Initial robo advisor

" Define an Amazon Lex bot with a single intent that establishes a conversation about requirements to suggest an investment portfolio for retirement. "

<https://user-images.githubusercontent.com/80833988/124681906-6bc12e00-de7e-11eb-98b9-d48959155925.mov>

### Enhanced Robo Advisor

" Create an Amazon Lambda function that validates the user's input and returns the investment portfolio recommendation. This includes testing the Amazon Lambda function and integrating it with the bot. "

<https://user-images.githubusercontent.com/80833988/124681923-77acf000-de7e-11eb-8e8a-9eaf4253ffd6.mov>

## Helps recruiters

The project was created in collaboration with Berkeley Fintech Bootcamp team

## License

MIT

☐ Contact me: ☐ nataliaburrey@gmail.com