Implementation of a robust, reliable and scalable AI ecosystem using Amazon Web Services (AWS).

Module Assignment for

CS5024 - Theory and Practice of Advanced AI Ecosystems

Student Name: **Akshata Bheemasamudra Mallikarjunappa**

Student ID: **23164204**

Revision Timestamp: 03/05/2024 06:09:17

Table of Contents

Abstract

A travel & hotel chain management company located in the region of Ireland that handles transport and hotel bookings and it also manages hotel administration, marketing and promotion. This company is tied up with multiple hotels, that span across the country, to provide different types of lodging facilities. The hotels typically offer different types of properties for travellers and tourists to stay. Resource management has become tedious due to customer churns, change in preferences, availability of resources, economic factors etc are some of the stumbling blocks while also trying to maintain the brand name since any small issue can quickly compound and impact brand image as a whole on a large scale.

In order to manage and allocate resources efficiently, for example, organizing bookings, accessing information in real-time, management complexities while also satisfying customer needs is crucial to successfully run hotels chains. Well-known brands, their franchises and other chains can easily manage their bookings & provide better customer service when all of their data is at one place which can be deployed on cloud and can be handled from here.

This model will predict what type of accommodation/lodging the different kind of customers are likely to choose to ensure its availability and correct allocation and also to attract customers with discounts.

This project uses AWS cloud services to,

1. *Deploy a cloud-based multi-property management system.*
2. *Deploy a Machine Learning model to predict hotel type a customer is likely to opt.*
3. *Scale the deployed cloud architecture to make it a robust and reliable ecosystem the business can use with the help of AWS services.*

Introduction

Business Goal: To optimize hotel resource allocation and reduce customer churn by giving discounts.

A hotel chain management wants to anticipate demand for different types of accommodation to help optimize inventory and pricing strategies in order to reduce customer churns and enhance overall customer experience by providing suitable accommodations, personalized amenities and improve satisfaction levels.

Machine Learning Goal: A multi-class classification problem containing 8 different classes from a historical customer data where the ML model needs to predict the type of hotel likely to be rented.

Dataset: Historical customer data which has categorical and numerical entries and contains the following features: Trip ID, Destination, Start date, End date, Duration In days, Traveler name, Traveler age, Traveler gender, Traveler nationality, Accommodation cost, Transportation type, Transportation cost & Accommodation type.

Key Performance Indicators: in functional language, % increase in customers booking their choice of hotels or % reduction in filing an inconvenience complaint. In non-functional language (confusion matrix & classification report), number of correctly predicted values in test cases.

Machine Learning Pipeline:

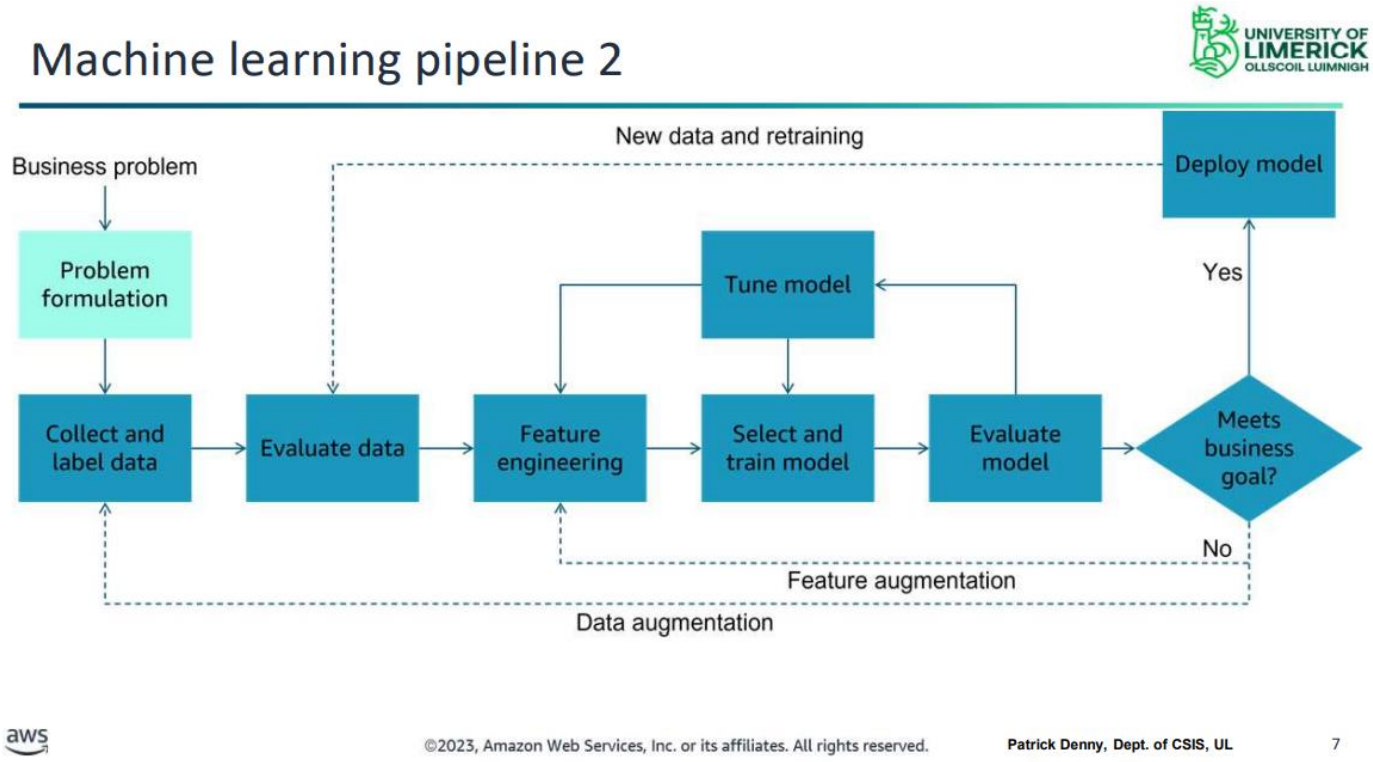

Figure 1: Machine Learning Pipeline

Figure 1 shows the Machine learning life cycle in AI ecosystem. As mentioned in Introduction section, the problem formulation has been discussed. Next stage is Collecting and labelling data and, in this project, it is assumed to have been collected from user activity and interaction through the company's website which is stored in cloud, Amazon RDS. Data is evaluated and feature engineering is performed in SageMaker's Jupyter Notebook which runs on EC2 instance inside a private subnet on a VPC. Selecting and training the model refers to selecting the right algorithm, hyperparameter and parameter. The tuning job uses XGBoost built-in algorithm provided by AWS Sagemaker to train ML model. Once the model is obtained, it is evaluated using various key performance indicators (as discussed above) and checks if it meets the business goal: accurately predicting the type of hotel that is most likely to be booked by a customer. If the model performs accurately, it is deployed for real-time use case else it will lead back to feature engineering step of the entire process and tuned again.

AI Ecosystem Architecture Used

There are three ways to interact and connect different components in AWS AI ecosystem: through software development keys, through command line interface and the last option is through *AWS management console* which is used in this project by creating AWS Free Tier account, to create S3 buckets, create lambda functions, make predictions, create API gateway and connect to lambda function.

The code implements the following basic architecture mentioned in Figure 2, however, the scalability section of the report discusses further on how this architecture can be made robust, more reliable and

scalable to ensure high availability, low latency and secure transaction while also backing up data and reducing computation time while training different models.

This AWS ecosystem implements a single tier architecture where the hotel chain management company has an account on the AWS cloud on which their website is hosted through the *client application*. Users are the customers who interact with the hotel management website and their data is collected and stored in *RDS* which is a relational database used to store configuration data. This is in the *eu-west-1 region*, Ireland. Using *Sagemaker* from this region, we need to acknowledge that SageMaker (is a fully managed service that is used to build, train and deploy machine learning models quickly) is hosted inside a *VPC* which has a *private subnet* on which an *EC2 instance* is running and the Sagemaker service is hosted on this instance. Sagemaker Notebook instances are fully managed ML compute instances running the Jupyter Notebook application. The *S3* bucket was manually created on the AWS management console using S3 service. We have used Jupyter notebook to prepare and process the data, written code to move the pre-processed data into the S3 bucket created, trained the model using sagemaker's built-in algorithm XGBoost, evaluated and deployed models in Amazon SageMaker. An endpoint is created and API gateway is used to make the predictions. Along with this, *Billing and cost management* service is used to to keep a track of cost and create a budget alert which has to be checked manually from the console. This is achieved with the help of *Amazon Key Management Service* that encrypts data and *CloudWatch* service which logs all the activities.

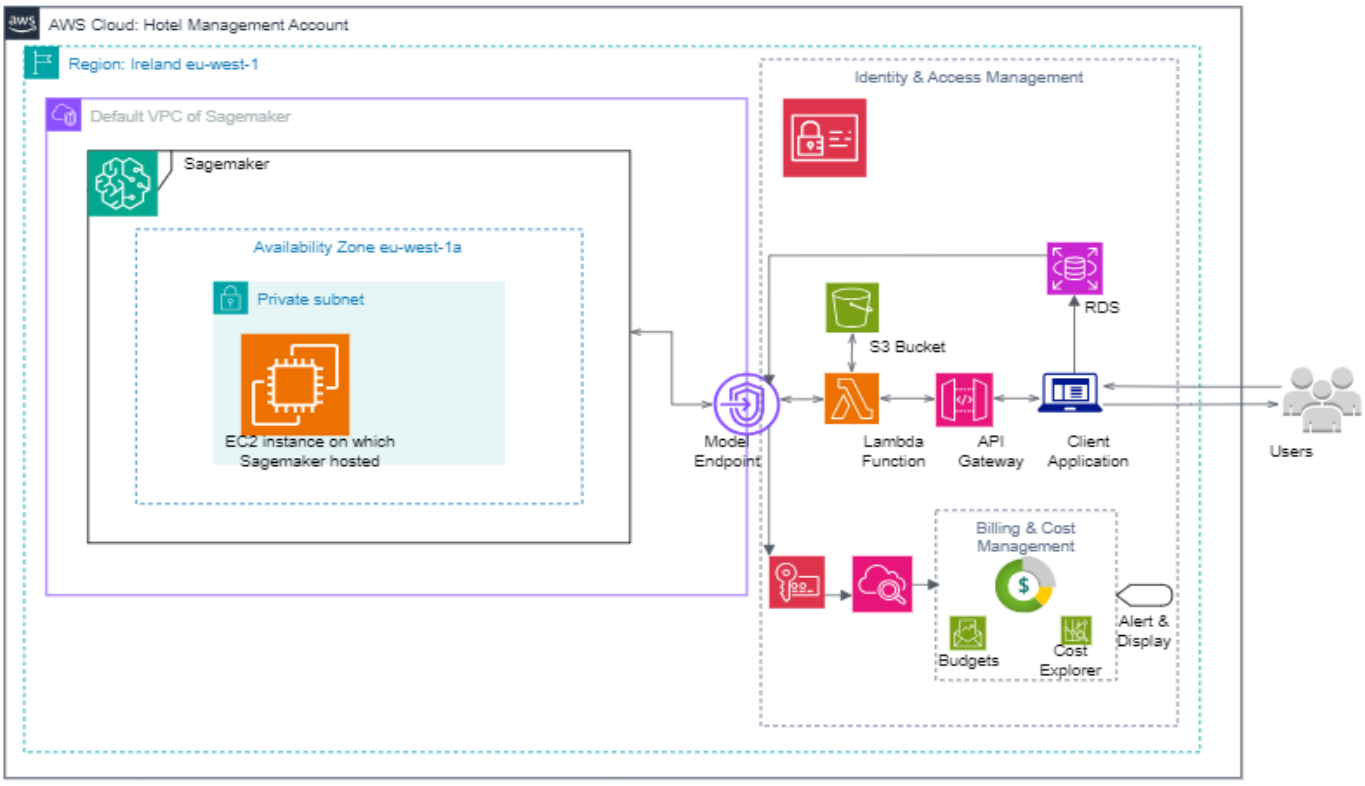

Figure 1: Architecture used to create AI/ML ecosystem for Hotel chain management model

Model Description

The model is hosted on Sagemaker where the data is cleaned, processed, trained and deployed which creates he model endpoint. The predictions are made on the test data, the accuracy was printed using confusion matrix which was upto 92%. As discussed earlier the key performance indicators are accuracy, precision, recall and f1 score. The model along with the predictions are saved in S3 bucket and further reaches the client application through lambda function and API gateway.

Scalability Considerations

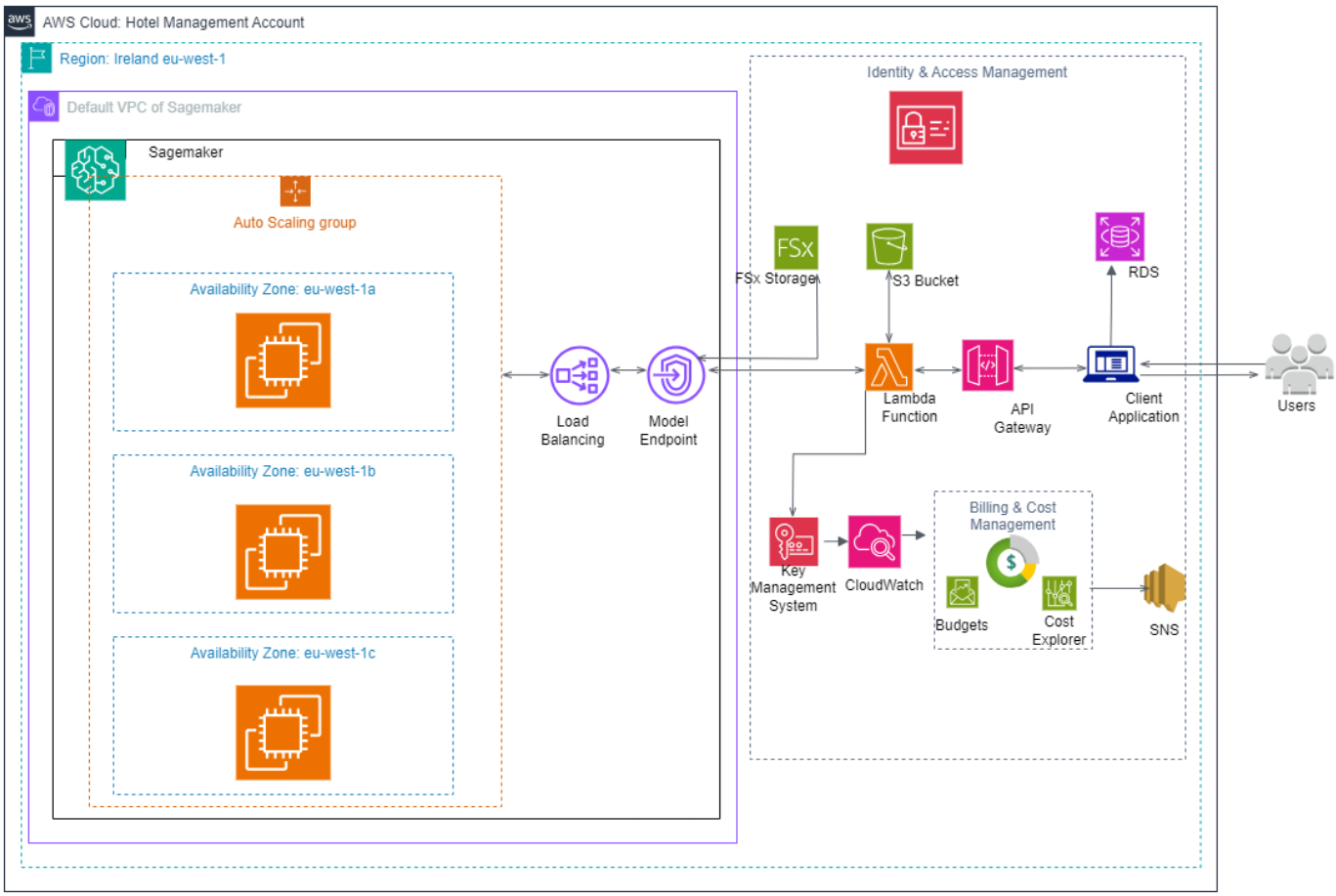

Figure 3: Scalable AI Ecosystem

The initial implementation is a single tier deployment architecture in the cloud. However, we see multiple issues as there is no reliability or scalability and high availability included. The new architecture makes use of autoscaling feature which deploys multiple EC2 instances in different availability zone as shown in Figure 3 (eu-west-1a, eu-west-1b, eu-west-1c) instead of only one zone which makes it high reliable and available in the event of compute failure or other factors which further introduces load balancing to manage multiple instances. This implementation also makes use of FSx storage instead of S3 by copying the data from S3 as it significantly reduces the need to perform Put and Get requests from S3 every single time a small change is made in the model, or while re- training, instead they can directly access from FSx which makes the computation between multiple data scientists easier and faster. This means S3 can then be used as a backup of data in the

event FSx fails to operate. This architecture further implements Amazon SNS to notify the architects about the cost and billing management w.r.t their preferences which helps reduce costs and manage resources accordingly.

The below figure 4 shows a snippet of the hotel-management model which was created, trained and deployed.

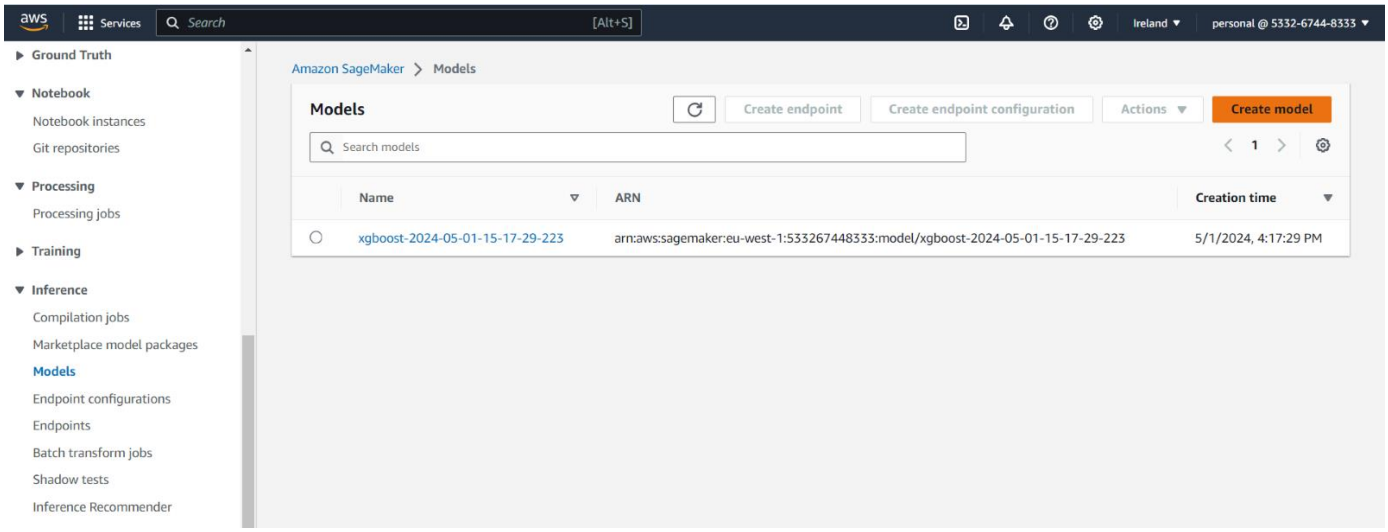

Figure 4: Snipper of Model creation from AWS management console

References

Dataset: Kaggle Dataset

Autoscaling Architecture: Autoscaling

SageMaker Deployment Tutorial: Youtube

XGBoost Algorithm: SageMaker Documentation

AWS Documentation: AWS Documentation

Figure 1: Patrick Denny. (2024, February 26). Introducing Amazon SageMaker - *Tuesday Week 10 Course Material*. Retrieved from CS5024 - Theory and Practice of Advanced AI Ecosystems: Link to PDF