Digital Egypt Pioneers Initiative

AWS Machine Learning Engineer Track

Text Classification using Amazon SageMaker Capstone Project Report

SUBMITTED TO: **DEPI's Committee**

SUBMITTED BY: **Abdullah Alsayed (21042655)**

Mohamad Ashour (21002520)

Muhammed Tariq Aboseif (21025242)

Raafat Elrais (21000337)

Oct - 2024

Abstract

This project presents an end-to-end text classification system built using Amazon Web Services (AWS). The system classifies BBC news articles into predefined categories utilizing a range of AWS services including Amazon S3, SageMaker, EC2, Comprehend, and Lambda. Key tasks include data preprocessing, feature extraction using natural language processing (NLP) techniques, model training, and deployment of the classification model. The model is trained using the BlazingText and Logistic Regression algorithms on SageMaker, with experiments tracked via MLflow. The deployed model handles real-time inference through a serverless AWS Lambda function, offering a scalable and efficient solution for text classification. The architecture demonstrates the seamless integration of cloud services, ensuring high performance and ease of management.

Table of Content

1. Project Overview

This project implements an end-to-end text classification system using AWS services, with a specific focus on classifying BBC news articles into predefined categories. The project utilizes various Amazon Web Services (AWS) for data processing, feature extraction, model training, and deployment. The primary goal is to demonstrate a robust and scalable solution for natural language processing (NLP) tasks using cloud-based machine learning infrastructure.

2. Objective

The main objective is to design and deploy a machine learning model that can accurately classify text data into categories using AWS SageMaker. The system leverages the full stack of AWS tools, focusing on scalability, ease of management, and deployment of the trained model for real-time inference.

3. System Architecture

The system architecture designed for this project utilizes AWS services to build a scalable and efficient solution.

3.1. Key components of the architecture:

1. Data Storage and Preprocessing:

    - AWS S3: Used to store raw BBC news articles, preprocessed data, and trained models.

    - AWS EC2/SageMaker Notebooks: Hosts the development environment for data preprocessing and cleaning. Python scripts are used to preprocess the dataset.

2. Feature Extraction:

    - AWS Comprehend: Used for extracting NLP features from the data, including entity recognition, key phrase extraction, and sentiment analysis.

3. Model Development:

- Amazon SageMaker: The BlazingText algorithm is used to train the text classification model.

- MLflow: Tracks experiments, hyperparameters, and model versions for better model management.

4. Model Deployment:

- The trained model is deployed to a SageMaker endpoint for real-time inference.

- AWS Lambda: Handles real-time inference requests with serverless architecture.

5. Pipeline Integration:

- AWS Step Functions: Can be used to orchestrate the entire workflow

- AWS CloudWatch: Monitors and logs the events across different stages of the deployed services.

4. Key AWS Services Utilized

- **Amazon S3:** Serves as the central storage for datasets and models.

- **Amazon EC2/SageMaker Notebooks:** For running preprocessing and model development.

- **AWS Comprehend:** Extracts critical NLP features.

- **Amazon SageMaker:** The core service for training and deploying the text classification model.

- **AWS Lambda:** Manages serverless, real-time inference requests.

- **MLflow:** Tracks experiments and manages version control of models.

- **AWS IAM:** Manages user access and permissions.

- **AWS SDK (Boto3):** Programmatic interaction with AWS services.

5. Project Workflow

5.1. Week 1: Setup and Data Preparation

- Tasks:

- Set up AWS environment including S3, EC2, and SageMaker.
- Data collection from BBC news articles.
- Data preprocessing using Pandas and NLTK libraries in Python.

- Deliverables:

- Configured AWS environment and preprocessed text dataset.

5.2. Week 2: Model Development

- Tasks:

- Feature extraction using AWS Comprehend.
- Model training using AWS SageMaker's BlazingText algorithm.

- Deliverables:

- Trained text classification model with evaluation results.

5.3. Week 3: Pipeline Integration

- Tasks:

- Model deployment on AWS SageMaker endpoint.
- Real-time inference using AWS Lambda.
- Versioning and experiment tracking with MLflow.

- Deliverables:

- Fully integrated text classification pipeline.

5.4. Week 4: Final Review and Presentation

- Tasks:

- End-to-end testing and validation of the system.
- Final project report and presentation.

- Deliverables:

- Complete documentation of the system and live demo of the deployed solution.

6. Model Evaluation

The text classification model was evaluated using standard metrics such as accuracy, precision, recall, and F1-score. The performance of the BlazingText algorithm was optimized through hyperparameter tuning using SageMaker's built-in tools.

7. AWS Services in Action

7.1. Setup S3 and EC2

Amazon S3 is used for data storage, including the uploading of raw and cleaned BBC news articles. EC2 instances are launched to serve as compute resources during data preprocessing.

7.1.1. Code Highlight:

```
s3 = boto3.client('s3')

bucket_name = 'my-text-classification-data'

s3.create_bucket(Bucket=bucket_name)
```

7.2. Data Cleaning and Preprocessing

Data cleaning is performed using Pandas for duplicate removal and text preprocessing (e.g., stopword removal and lemmatization). This process prepares the data for machine learning algorithms.

7.2.1 Code Highlight:

```
df.drop_duplicates(keep="first", inplace=True)

df["data"] = df["data"].apply(preprocessing)
```

7.3. Model Training with Amazon SageMaker

Amazon SageMaker is used to train the text classification model with Logistic Regression. The training and testing data are stored in S3 and then loaded into SageMaker. The model is trained on scalable, cloud-based infrastructure.

7.3.1 Code Highlight:

```
lr_estimator = sagemaker.estimator.Estimator(
    container, role, instance_count=1, instance_type='ml.c5.2xlarge',
    output_path=f's3://{bucket}/{prefix}/output',
    sagemaker_session=session
)
```

7.4. NLP Feature Extraction with Amazon Comprehend

AWS Comprehend is employed for entity recognition, sentiment analysis, and key phrase extraction from the dataset. These extracted

features are saved back to S3 for model training.

7.4.1Code Highlight:

```
comprehend.detect_entities(Text=chunk, LanguageCode='en')
```

7.5. Experiment Tracking and Model Deployment

MLflow is used for experiment tracking and managing model versioning. The trained Logistic Regession model is logged and deployed via SageMaker for real-time inference.

7.5.1Code Highlight:

```
mlflow.log_params(hyperparameters)

mlflow.sagemaker.log_model(lr_estimator.model_data, "logistic-regression-model")
```

7.6. Real-Time Inference with AWS Lambda

A serverless inference endpoint is created using AWS Lambda, which allows real-time text classification requests
to be processed via the deployed SageMaker model.

7.6.1. Code Highlight:

```
lambda_client.create_function(
   FunctionName='BBCTextClassifier', Handler='lambda_function.lambda_handler',
)
```

8. Conclusion

The project successfully implements a scalable and robust text classification system using Amazon SageMaker and other AWS tools. It demonstrates proficiency in handling large-scale NLP tasks and cloud-based machine learning model deployment. This end-to-end solution showcases the integration of various AWS services, from data storage to real-time inference, making it suitable for practical applications in text classification tasks.