Determining job category using AWS machine learning cloud services

https://doi.org/10.31713/MCIT.2023.070

Vitalii Pavliuk
Department of Automation, Electrical Engineering and Computer-Integrated Technologies
National University of Water and Environmental Engineering
Rivne, Ukraine

Volodymyr Drevetskyi
Doctor of technical sciences, Professor
National University of Water and Environmental Engineering
Rivne, Ukraine
v.v.drevetskyi@nuwm.edu.ua

Abstract—**In this article, the focus is on the modern role of information technology in the labor market, with special emphasis on automation and optimization of processes through cloud solutions. The main theme is the creation of a system for the automatic categorization of job vacancies using Amazon Web Services (AWS). The authors describe the key advantages of implementing cloud technologies, such as scalability, cost savings, availability, and high productivity. Specific AWS services are then discussed, including ECS, DynamoDB, AWS Glue, Amazon SageMaker, and AWS Lambda, which can be applied for the implementation of this system. The article aims to show how AWS technologies can be used to improve efficiency.**

Keywords—information technology; cloud services; Amazon Web Services; job categorization; automation; labor market.

I. Introduction

In today's world, it's impossible to overstate the role of information technology, which has become an integral component of various aspects of our lives. Particularly relevant is the implementation of IT in the labor market, where technology opens up endless opportunities for automation and optimization of processes aimed at increasing efficiency and productivity. One of the main advantages of using information technology is the ability to apply cloud solutions. Utilizing cloud technologies offers several benefits, including:

- Scalability: Cloud services allow for easy scaling of resources according to needs, optimizing computational power.
- Cost Savings: The absence of the need to purchase and maintain one's own equipment reduces expenses, allowing you to rent resources only when necessary.
- Availability: Cloud solutions provide access to data and resources from anywhere in the world, contributing to the flexibility and mobility of work processes.
- High Productivity: The ability to rent high-performance computing machines for a short term allows for complex calculations and data analysis in a short amount of time.

Within the framework of this article, we explore the creation of a system for automatic job categorization using Amazon Web Services (AWS). Utilizing AWS allows us to take advantage of all the benefits of cloud technologies, facilitating an effective and flexible project implementation. We will be using AWS services such as ECS, DynamoDB, AWS Glue, Amazon SageMaker, and AWS Lambda. These tools will not only aid in automating the job categorization process but also ensure stable and efficient system operation at various stages of its implementation.

II. Comparing Cloud Computing Services

Cloud computing has transformed the way organizations and individual users interact with technology and process data. Today's major cloud service providers are Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

AWS is the most mature and widely used provider globally. It offers a wide range of services and tools that can meet the needs of virtually any business or project. In terms of reliability, AWS offers advanced solutions for backup, recovery, and security.

Microsoft Azure, on the other hand, is often chosen by companies already using Microsoft products like Windows Server, Active Directory, and others. Azure offers deep integration with Microsoft products and has a wide array of supported services.

Google Cloud Platform (GCP) is known for its capabilities in big data processing and machine learning. It also offers efficient solutions for distributed computing and data storage.

Thus, considering the experience, diversity of services, and global presence, AWS stands out as a highly effective solution for cloud computing. This makes AWS an attractive choice for organizations looking to scale their operations, optimize resources, and increase productivity through cloud technologies. In this article, we will also be using Amazon Web Services as the cloud computing provider.

III. Data Collection

To develop and train a job category prediction model, a large amount of data is required. In this article, the data source is the jobs.dou.ua website, one of the most popular job search resources in the IT industry in Ukraine. Web scraping is used for automatic data collection from web pages. Selenium library for Python is used to scrape job vacancies from jobs.dou.ua. The script, written with Selenium, visits the job pages, gathers information on each job (like title, description, category, location, etc.), and saves it.

This script is deployed using AWS Elastic Container Service (ECS), a powerful cloud tool for container management. AWS ECS allows easy definition of virtual machine specs in the cloud using a Docker file. AWS ECS manages containers at a high level, optimizing resource allocation and ensuring high availability.

After the data collection stage, it's essential to save the data in a structured and optimized format. AWS DynamoDB is used for data storage, a high-availability NoSQL database that's ideal for fast access to large data sets. DynamoDB's deep integration with the AWS ecosystem simplifies interaction with other cloud services.

To automate data storage, the article uses integration of the Selenium script with AWS SDK. This allows automatic transmission of collected data to a DynamoDB table right after collection. Going forward, the collected data will be used for export, cleansing, and preparation processes, as well as training a machine learning model aimed at labor market analysis and classification.

IV. EXPORT AND DATA PREPARATION

After the stages of gathering and storing data in DynamoDB, the next step involves exporting this data for further preparation and processing. To create aggregated files containing all the collected job vacancies along with their respective categories, it's essential to use effective data processing tools.

For this task, we use AWS Glue to export the data from DynamoDB to Amazon S3. AWS Glue is a scalable and fully managed ETL (Extract, Transform, Load) service, which significantly simplifies the data preparation process for analytical needs.

We create a task in AWS Glue with the aim to read data from the relevant DynamoDB table, process it into the required format, and store the results in an Amazon S3 bucket in CSV format. This allows us to have a centralized and optimized data repository, ready for further analytical operations and machine learning model processing. The task also splits the data into training and validation sets, so we end up with two files.

After creating the ETL task, it needs to be run. AWS Glue automatically provides the necessary resources for task execution and scales according to the volume of data. After the successful execution of the task, the data from DynamoDB has been exported to the Amazon S3 bucket as structured files, ready for further processing and analysis. This process is an essential part of data preparation, as it ensures the uniformity, structure, and availability of data for subsequent stages of work with machine learning.

V. MACHINE LEARNING MODEL

Amazon SageMaker is a fully-featured service that enables developers and researchers to quickly build, train, and deploy machine learning models at any scale. SageMaker includes a wide range of capabilities, from convenient notebooks for model development and integration with popular frameworks (such as TensorFlow and PyTorch), to automatic scaling and resource management for training and deploying models.

In addition, Amazon SageMaker offers an extensive collection of built-in algorithms that cover a broad spectrum of machine learning tasks, from regression and classification to neural networks and model complexity enhancement. These algorithms are optimized for high-performance execution on large-scale data and are designed to work 'out-of-the-box' without the need for complex configurations.

Given this diversity and flexibility that Amazon SageMaker provides, we had to make a well-founded choice of algorithm for our job classification task. Our choice was the BERT algorithm (Bidirectional Encoder Representations from Transformers). BERT is a modern natural language processing algorithm, known for its ability to effectively analyze text, taking into account the context of words in a sentence. Using BERT with SageMaker gives you access to a high-quality algorithm, plus the scalability and optimization that comes with AWS's cloud infrastructure. It's essential to understand why we went for BERT:

- Understanding Context: Unlike traditional language models, BERT actually learns the context of a word by peeping at the words before and after it. This is a big deal, especially because job listings often contain highly contextual information.
- High Accuracy: BERT has been like the Michael Jordan of natural language processing tasks, showing top-level accuracy in things like text classification. This can help us achieve high accuracy in job vacancy categorization.
- Pre-Trained Models: BERT brings a treasure trove of pre-trained models built on massive datasets. It's like tapping into an existing reservoir of wisdom to elevate our model, without having to train it from scratch.
- Transfer Learning: Thanks to BERT, we can use transfer learning where the core model can be fine-tuned to specific job vacancy data, allowing us to embed the unique features and context of our domain.
- Speed and Efficiency: Despite the complexity of the model, BERT is optimized for quick processing, making for efficient training and high-speed predictions.
- Language Flexibility: BERT has been trained on datasets in multiple languages, making it effective for handling texts in various languages.
- Adaptation to Complex Tasks: BERT has a deep architecture that enables it to adapt effectively to complex tasks and data structures.

All in all, given these advantages, BERT is a great choice for our job classification task. It promises deep text understanding and high prediction accuracy.

For effective development and tuning of the machine learning model, we'll set up a notebook in Amazon SageMaker. A notebook is like your wizard's spellbook for coding, parameter tweaking, and data experimenting.

After configuring the parameters and choosing the BERT algorithm in the SageMaker notebook, we initiate the training process by pointing to our data stored in Amazon S3. SageMaker takes care of the resource provisioning like a good backstage manager. Using a test dataset, we'll evaluate the model's performance to see how well it's playing the job-classification tune. Once we're past the training and evaluation stage, we can deploy the model via Amazon SageMaker. The service allows you to launch the model as a web service that can accept requests and return real-time predictions.

VI. Web service using AWS Lambda

After deploying the machine learning model on Amazon SageMaker, the next step is to integrate this model with a web interface that allows users to get predictions through HTTP requests. AWS Lambda and Amazon API Gateway play key roles in this process.

Initially, we create an AWS Lambda function via the AWS Management Console. The essence of the Lambda function is that it receives HTTP requests, interacts with the machine learning model in SageMaker, and sends back predictions. It's crucial to set up this function correctly so that it can accept and process data in a specific format and also send back responses in a user-friendly format.

To let the Lambda function communicate with SageMaker and other AWS services, we need to create an IAM role with appropriate access policies. This gives the Lambda function the needed permissions to call the SageMaker model and handle data.

To enable the Lambda function to receive HTTP requests from the Internet, we should integrate it with Amazon API Gateway. API Gateway lets us create, publish, and manage APIs in a scalable environment. After configuring the API Gateway, we deploy the API, which generates a public URL for interacting with the Lambda function.

With the configuration and deployment of the API Gateway complete, we can proceed to test the web service by sending HTTP requests to the public URL generated via API Gateway. This can be done using various tools like Postman or CURL, as well as by integrating the API into web or mobile apps for real-world use.

Conclusion

This article explored the process of gathering, processing, and analyzing job vacancies from the website jobs.dou.ua using various AWS services. It demonstrated how cloud computing technologies can be used to create a complex data processing system and develop a machine learning model.

The main stages of work included:

- Data Collection - Using Selenium, a script was created for automatic collection of job vacancies from jobs.dou.ua.
- Data Storage - The collected data was stored in an Amazon DynamoDB database, providing a fast and reliable way to stash the goods.
- Data Export - With the help of AWS Glue, the data was exported from DynamoDB to Amazon S3 for further analysis.
- Machine Learning Model Development - Using Amazon SageMaker, a machine learning model was developed to predict job categories.
- Web Service Implementation - Utilizing AWS Lambda and API Gateway, a web service was set up to let users fetch those model predictions through HTTP requests.

For future work, we can gather data from other job search websites that don't have clear categories, and now we can predict these categories using the machine learning model we've built.

Overall, this article showcased the potential of cloud technologies in the realm of data processing and machine learning. Using AWS allowed us to automate and optimize various aspects of the process, leveling up the efficiency and flexibility of the whole shebang.

References

[1] “AWS Well-Architected Framework.” [Online]. Available: https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html. [Accessed: October 17, 2023].

[2] T. Nguyen, *Gentle Introduction to How AWS ECS Works with Example Tutorial*. [Online]. Available: https://medium.com/boltops/gentle-introduction-to-how-aws-ecs-works-with-example-tutorial-cea3d27ce63d. [Accessed: October 17, 2023].

[3] “Build, Train, and Deploy a Machine Learning Model with Amazon SageMaker.” [Online]. Available: https://aws.amazon.com/getting-started/hands-on/build-train-deploy-machine-learning-model-sagemaker/. [Accessed: October 17, 2023].

[4] “Introduction to AWS Lambda Serverless Architecture.” [Online]. Available: https://www.serverless.com/learn/quick-start/. [Accessed: October 17, 2023].