ILLINOIS TECH

Project Proposal

CSP 554: Big Data Technologies

By

Name	A_ID	Email
Divyansh Soni	A20517331	dsoni2@hawk.iit.edu

Under the Guidance of Prof. Joseph Rosen

1 Introduction to the ML Use Case

In this section, we introduce sentiment analysis for product reviews using a multi-class classification approach. We have selected this specific use case to demonstrate the construction of a big data machine learning system on Amazon SageMaker.

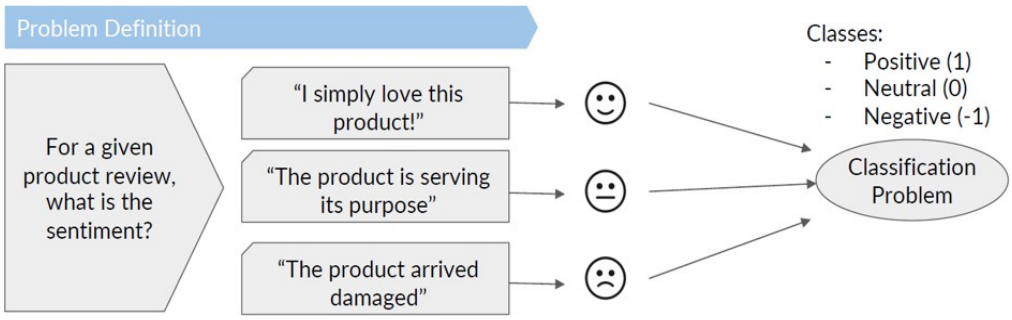


Figure 1.1 Use Case: sentiment analysis for product reviews

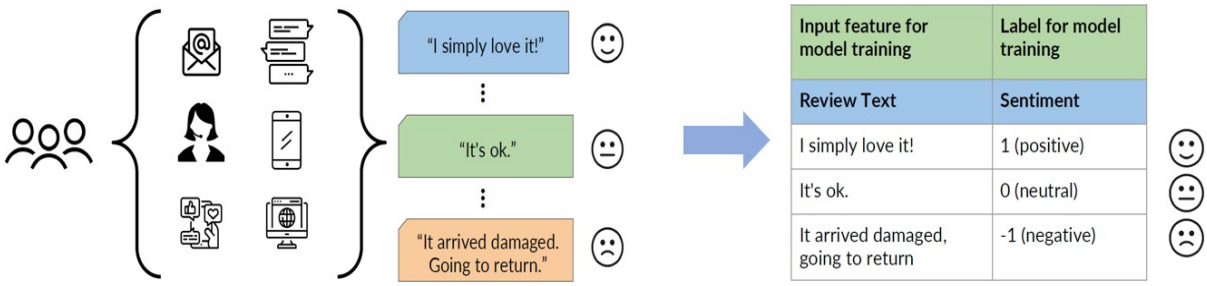


Figure 1.2 Sentiment Analysis for Product Reviews

2 Big Data Machine Learning System on AWS Sagemaker

In this section, we will delve into the complexities and solutions of constructing a Big Data Machine Learning system using AWS SageMaker. We begin by researching the industrial challenges that practitioners encounter in this context, focusing on the intricacies of pipeline orchestration and automation. We will explore the integration of AutoML into the Big Data ML workflow, aiming to address the challenges. We introduce AutoML, its benefits, and how it

streamlines the development process in a big data machine learning system. Additionally, we delve into SageMaker Pipelines and its role in the broader machine-learning workflow. Finally, we introduce SageMaker Autopilot, Amazon SageMaker's unique approach to automated machine learning (AutoML). We illustrate how SageMaker Autopilot seamlessly integrates AutoML and supports the development of a big data system for product review sentiment analysis. This section provides a comprehensive understanding of the tools and techniques essential for constructing an efficient Big Data Machine Learning system on AWS SageMaker.

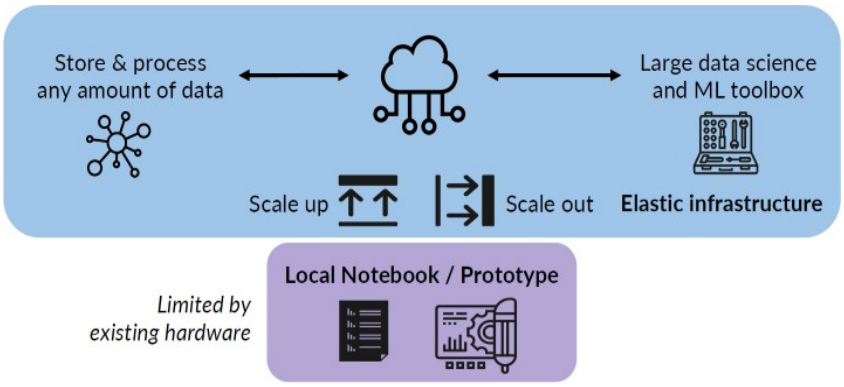
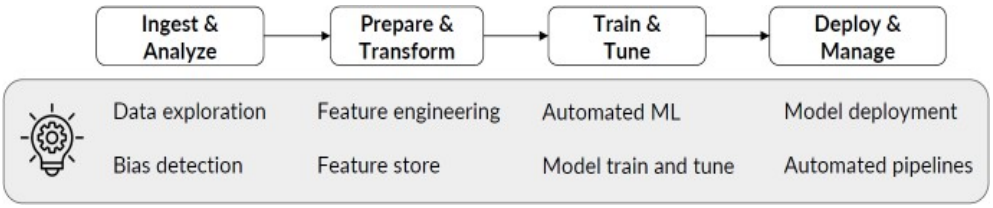


Figure 2.1 Machine Learning in Cloud

By migrating the big data machine learning system into the Cloud, we are no longer bound by resource limitations, such as the laptop's CPU processing power or memory. We can run data analysis on virtually any size of data. We can slice the dataset and run data transformations in parallel. We can switch from CPU to GPU if we want to speed up the model training.



1.3 Pipelines in Big Data Machine Learning System

2.1 Navigating the Complexities: Challenges in Constructing Big Data Machine Learning Systems

In this section, we will research and explore the industrial challenges encountered when constructing Big Data Machine Learning systems, with a particular focus on the complexities of pipeline orchestration and automation. These challenges encompass extended development timelines, the demand for specialized skills, and resource constraints that industry practitioners often grapple with.

2.2 Leveraging AutoML in the Big Data ML Workflow

In this section, we turn our attention to the solutions that can address the challenges highlighted in Section 2.1. We start by introducing AutoML, its definition, and the benefits it brings, and big data machine learning system workflow with AutoML. We will also delve into the integration of AutoML into the big data machine learning system workflow, along with the specific tasks that automated machine learning is designed to handle. Our mission is to not only highlight challenges in building big data machine learning systems but also investigate how AutoML can offer innovative solutions in big data machine learning workflow such as streamline the development process.

2.2.1 Big Data ML System Workflow

In this section, we will provide an in-depth exploration of the machine-learning workflow, specifically focusing on machine-learning pipelines with SageMaker Pipelines. We will delve into how AutoML seamlessly integrates into the broader machine-learning workflow, offering insights into its unique advantages and contributions.

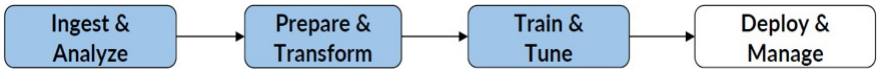


Figure 2.2.1 Big Data ML System Workflow

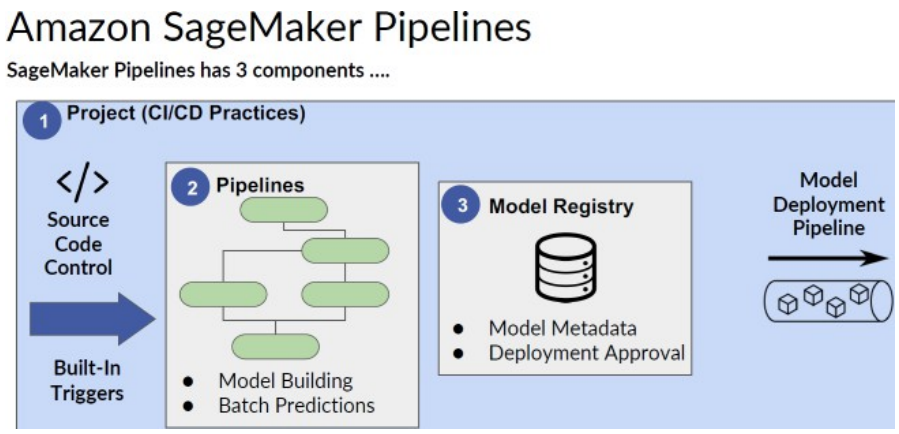


Figure 2.2.2 Amazon SageMaker Pipelines

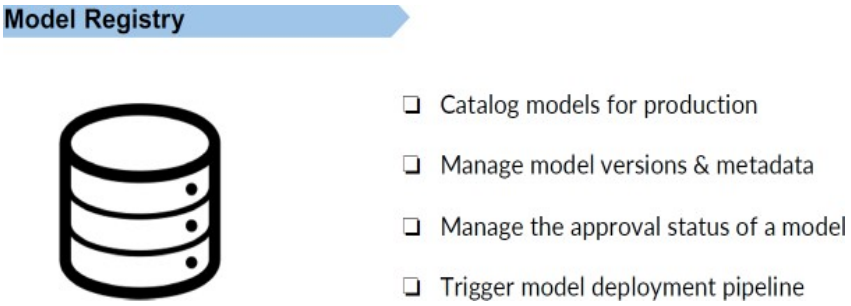


Figure 2.2.3 Model Registry

2.3 Unveiling SageMaker Autopilot: A Unique Approach to AutoML in Amazon SageMaker

In this section, we will introduce Amazon SageMaker's automated machine learning (AutoML) service, known as SageMaker Autopilot. We will delve into how SageMaker Autopilot uniquely integrates AutoML and supports the development of a big data system for product review sentiment analysis.

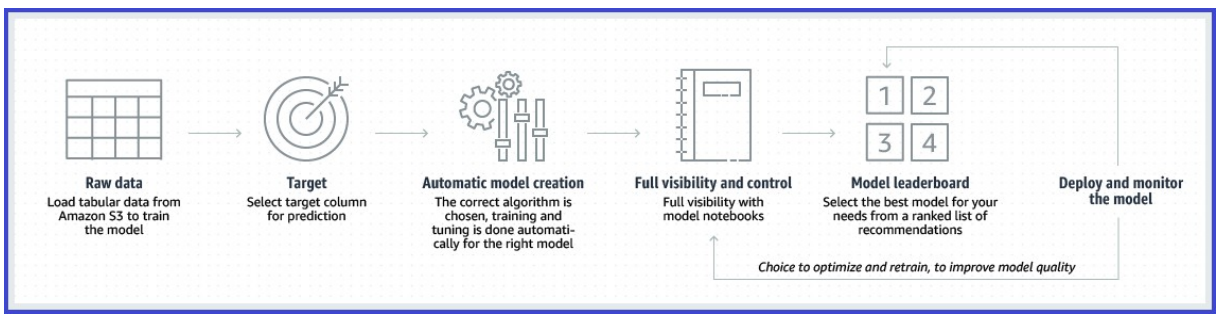


Figure 2.3.1 AutoML Process Managed by Autopilot

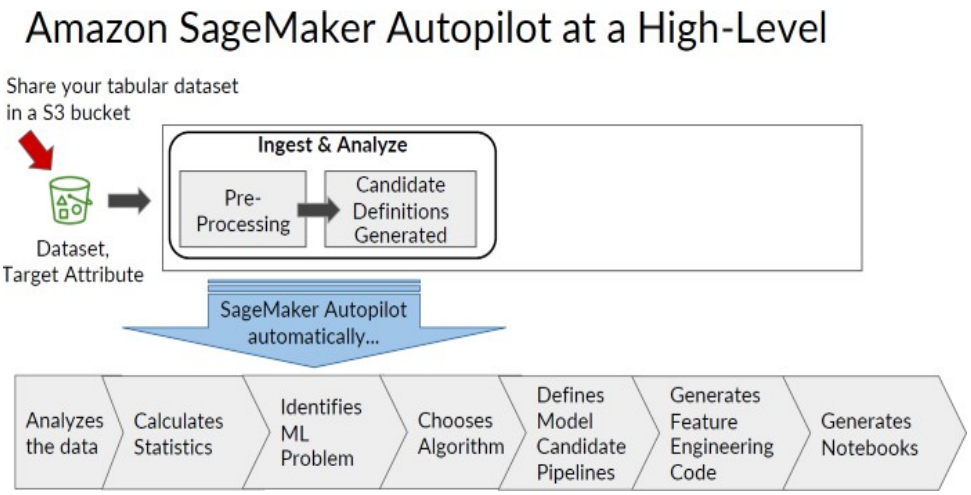


Figure 2.3.2 Amazon SageMaker Autopilot at a High Level

3 Data Preparation and Transformation on SageMaker

In this section, we delve into the vital steps of data preparation and transformation using Amazon SageMaker. Our goal is to optimize model performance, ensure fairness, and facilitate data-driven decisions in building the product review big data machine learning system. We introduce feature engineering, covering feature selection, creation, and transformation, and demonstrate how these engineered features are stored in a centralized feature store (Amazon SageMaker Feature Store). We'll also discuss the integration of the BERT NLP model to scale feature engineering within our big data machine learning system, providing a comprehensive understanding of our data preparation process.

Divyansh Soni, 11/9/2023:
After the discussion with Prof. Rosen, the code implementation in section 3 can be done by Spark, I will modify the description of the big data system architecture and update this section after the proposal submission. If time allows, we will use EMR Spark VM to do data profiling, not pandas, numpy, etc.

3.1 Introduction to Bias, Feature Importance

In this section, we introduce the fundamental concepts of bias, bias detection, and feature importance. We emphasize the importance of integrating these elements into our ML pipelines to ensure fairness and enhance model performance. We will elucidate the significance of bias detection and feature importance within the context of building machine learning pipelines for product review sentiment analysis. Furthermore, we will outline our approach to incorporating these concepts into our system using the SageMaker toolkit.

Related code implementation - preparing and transforming the data for model training (perform feature engineering).

Basic Toolkit & code implementation in the project:

- Amazon Simple Storage Service (S3): we'll build a data lake on top of S3
- AWS Data Wrangler: we'll use it to load or unload data from the data lake, especially use it to register the CSV data with the AWS Glue Data Catalog
- AWS Glue Data Catalog: we'll use it to register and catalog the data stored in S3
- Amazon Athena: we'll use it to query datasets stored in the data lake on S3
- Amazon SageMaker Clarify: we'll use it to detect statistical bias, and drift in data and model.
- Amazon SageMaker Data Wrangler: we'll use it to calculate feature importance on the product review data set

3.2 Data Profiling

In this section, we will embark on the process of statistical data analysis through visualizations. Visualizing data is a powerful tool for gaining insights, identifying patterns, biases, understanding feature importance, and making data-driven decisions. It also aids us in deciding how to approach feature engineering when building a big data machine learning system on Amazon SageMaker. We will delve into statistical analysis through data visualization in the context of our project on product review sentiment analysis.

4 Train and Tune Models on SageMaker

We will build a custom model using the BERT algorithm on SageMaker. Our focus in this section is exploring best practices for distributed model training and hyperparameter tuning. We will dive into advanced model deployment options, as well as learn methods that can be used to monitor models once they're deployed and perform data labeling at scale. We will take advantage of popular algorithms to optimize a machine-learning model through automated hyperparameter tuning and model selection.

4.1 Introduction to Automatic Model Tuning Algorithms

Hyperparameter tuning is usually a time-consuming and compute-intensive process when constructing a big data machine learning system. When training machine learning models on SageMaker, hyperparameter tuning represents a crucial step in achieving the highest quality model. This may entail obtaining a model with the highest achievable accuracy or the lowest attainable error. In this section, we will introduce a few popular algorithms used for automated model tuning and hyperparameter tuning on Amazon SageMaker.

In building this big data machine learning system, we avoid manually selecting hyperparameter values depending on the machine learning models that we choose for the use case: product review sentiment analysis. We leverage automatic model tuning to fine-tune the hyperparameters to find the best-performing values. The goal of using the algorithms for automatic model tuning is to fine-tune the hyperparameters to discover the most optimal values for performance.

4.2 Best Practices in Hyperparameter Tuning for Big Data ML Systems

In Section 4.2, we delve into the best practices for hyperparameter tuning in the context of developing big data machine learning systems. This section explores the intricacies of model evaluation, the resource-intensive nature of hyperparameter tuning, the significance of narrowing down the range of values, and the advantages of early stopping. We also discuss

the considerations for concurrent job execution in the pursuit of optimal outcomes. These insights will guide our approach to hyperparameter tuning and enhance the efficiency and effectiveness of machine learning endeavors.

In the development of a big data machine learning system, we evaluate the model's performance continuously during the model training process to find the model's accuracy using a holdout validation dataset. During this process, we fine-tune the model parameters and hyperparameters as necessary. 4Hyperparameter tuning is a resource-intensive process, both in terms of time and computation. The computational demands are directly correlated with the quantity of hyperparameters under consideration. While Amazon SageMaker enables us to optimize up to 20 different hyperparameters concurrently, it's generally more effective to work with a more modest subset of hyperparameters. Similarly, it's advisable to narrow down the range of values to investigate these hyperparameters. The selection of values for hyperparameters has a substantial impact on the success of the optimization process. While the temptation might be to specify an extensive range of values to explore all possibilities, superior results are often achieved by constraining our search to a more focused range of values. When we activate early stopping in the hyperparameter tuning job, the individual training jobs initiated by the tuning job can terminate prematurely if the objective metric ceases to show continuous improvement. This early cessation of individual training jobs accelerates the overall completion of the hyperparameter tuning job and reduces costs. The final best practice here involves using a limited number of concurrent training jobs.

SageMaker does offer the capability to run multiple jobs concurrently during the hyperparameter tuning process. While it might be tempting to use a higher number of concurrent jobs to expedite the tuning process in the context of big data machine learning system development, it's essential to recognize that the hyperparameter tuning process relies on previously completed training jobs to identify the optimal outcomes. In our big data machine learning system, we opt for a smaller number of concurrent jobs when executing the hyperparameter tuning job.

Reference: study and research resources

Libqjty, Edo, qt al. ‘Elastic Machinq Lqajning Algojithms in Amazon SagqMakqj’. Pjocqqdings of thq 2020 ACM SIGMOD Intqjnational Confqjqncq on Managqmqnt of Data, ACM, 2020, pp. 731–37. DOI.ojg (Cjossjqf), https://doi.ojg/10.1145/3318464.3386126.

PQJJonQ, ValQJio, Qt al. ‘Amazon SagQMakQJ Automatic ModQl Tuning: ScalablQ GJadiQnt-FJQQ Optimization’. PJocQQdings of thQ 27th ACM SIGKDD ConfQJQncQ on KnowlQdgQ DiscovQJy & Data Mining, ACM, 2021, pp. 3463–71. DOI.oJg (CJossJQf), https://doi.oJg/10.1145/3447548.3467098.

- AWS Data Wrangler
- AWS Glue
- Amazon Athena
- Matplotlib
- Seaborn
- Pandas
- Numpy
- Practical Data Science on the AWS Cloud
- Amazon SageMaker Autopilot
- RoBERTa: A Robustly Optimized BERT Pretraining Approach
- Fundamental Techniques of Feature Engineering for Machine Learning
- Amazon SageMaker Model Training (Developer Guide)
- Amazon SageMaker Debugger: A system for real-time insights into machine learning model training
- The science behind SageMaker’s cost-saving Debugger
- Amazon SageMaker Debugger (Developer Guide)
- Hugging Face open-source NLP transformers library
- RoBERTa model
- Women's E-Commerce Clothing Reviews (kaggle.com)