**5121-RANIPPETTAI ENGINEERING COLLEGE**

**E-COMMERCE APPLICATION ON IBM CLOUD FOUNDARY**

Phase-3

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**Create an ecommerce streaming pipeline**

## Objectives

* Validate incoming data and apply corrections to it where possible.
* Analyze clickstream data to keep a count of the number of views per product in a given time period. Store this information in a low-latency store. The application can then use the data to provide number of people who viewed this product messages to customers on the website.
* Use transaction data to inform inventory ordering:
  + Analyze transaction data to calculate the total number of sales for each item, both by store and globally, for a given period.
  + Analyze inventory data to calculate the incoming inventory for each item.
  + Pass this data to inventory systems on a continuous basis so it can be used for inventory purchasing decisions.
* Validate incoming data and apply corrections to it where possible. Write any uncorrectable data to a dead-letter queue for additional analysis and processing. Make a metric that represents the percentage of incoming data that gets sent to the dead-letter queue available for monitoring and alerting.
* Process all incoming data into a standard format and store it in a data warehouse to use for future analysis and visualization.
* Denormalize transaction data for in-store sales so that it can include information like the latitude and longitude of the store location. Provide the store information through a slowly changing table in BigQuery, using the store ID as a key.

### Data

The application processes the following types of data:

* Clickstream data being sent by online systems to Pub/Sub.
* Transaction data being sent by on-premises or software as a service (SaaS) systems to Pub/Sub.
* Stock data being sent by on-premises or SaaS systems to Pub/Sub.

**Note:** All data is sent as JSON. This format is normal for clickstream data but not necessarily for transaction or stock data.

### Task patterns

The application contains the following task patterns common to pipelines built with the Apache Beam SDK for Java:

* [Apache Beam schemas to work with structured data](https://cloud.google.com/dataflow/docs/tutorials/ecommerce-java#structured-data)
* [JsonToRow to convert JSON data](https://cloud.google.com/dataflow/docs/tutorials/ecommerce-java#json-to-row)
* [The AutoValue code generator to generate plain old Java objects (POJOs)](https://cloud.google.com/dataflow/docs/tutorials/ecommerce-java#use-autovalue)
* [Queuing unprocessable data for further analysis](https://cloud.google.com/dataflow/docs/tutorials/ecommerce-java#queue-unprocessable-data)
* [Serial data validation transforms](https://cloud.google.com/dataflow/docs/tutorials/ecommerce-java#data-validation-transforms)
* [DoFn.StartBundle to micro-batch calls to external services](https://cloud.google.com/dataflow/docs/tutorials/ecommerce-java#micro-batch-calls)
* [Side-input patterns](https://cloud.google.com/dataflow/docs/tutorials/ecommerce-java#side-input-pattern)

## Costs

In this document, you use the following billable components of Google Cloud:

* BigQuery
* Cloud Bigtable
* Cloud Scheduler
* Compute Engine
* Dataflow
* Pub/Sub

To generate a cost estimate based on your projected usage, use the [pricing calculator](https://cloud.google.com/products/calculator). New Google Cloud users might be eligible for a [free trial](https://cloud.google.com/free-trial).

When you finish the tasks that are described in this document, you can avoid continued billing by deleting the resources that you created. For more information, see [Clean up](https://cloud.google.com/dataflow/docs/tutorials/ecommerce-retail-pipeline#clean-up).

## Before you begin

1. Sign in to your Google Cloud account. If you're new to Google Cloud, [create an account](https://console.cloud.google.com/freetrial) to evaluate how our products perform in real-world scenarios. New customers also get $300 in free credits to run, test, and deploy workloads.
2. [Install](https://cloud.google.com/sdk/docs/install) the Google Cloud CLI.
3. To [initialize](https://cloud.google.com/sdk/docs/initializing) the gcloud CLI, run the following command:

gcloud init

1. Create or select a Google Cloud project.

**Note**: If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

* + Create a Google Cloud project:

gcloud projects create PROJECT\_ID

* + Select the Google Cloud project that you created:

gcloud config set project PROJECT\_ID

1. [Make sure that billing is enabled for your Google Cloud project](https://cloud.google.com/billing/docs/how-to/verify-billing-enabled#console).
2. Enable the Compute Engine, Dataflow, Pub/Sub, BigQuery, Bigtable, Bigtable Admin, and Cloud Scheduler APIs:
3. gcloud services enable compute.googleapis.com dataflow.googleapis.com pubsub.googleapis.com bigquery.googleapis.com bigtable.googleapis.com bigtableadmin.googleapis.com  cloudscheduler.googleapis.com
4. Create local authentication credentials for your Google Account:

gcloud auth application-default login

1. Grant roles to your Google Account. Run the following command once for each of the following IAM roles: roles/iam.serviceAccountUser

gcloud

Create a user-managed worker service account for your new pipeline and grant the necessary roles to the service account.

1. To create the service account, run the [gcloud iam service-accounts create](https://cloud.google.com/sdk/gcloud/reference/iam/service-accounts/create) command:

gcloud iam service-accounts create retailpipeline \  
    --description="Retail app data pipeline worker service account" \  
    --display-name="Retail app data pipeline access"

1. Grant roles to the service account. Run the following command once for each of the following IAM roles:
   * roles/dataflow.admin
   * roles/dataflow.worker
   * roles/pubsub.editor
   * roles/bigquery.dataEditor
   * roles/bigtable.admin

## Create the example sources and sinks

This section explains how to create the following:

* A Cloud Storage bucket to use as a temporary storage location
* Streaming data sources using Pub/Sub
* Datasets to load the data into BigQuery
* A Cloud Bigtable instance

### Create a Cloud Storage bucket

Begin by creating a Cloud Storage bucket using the Google Cloud console or Google Cloud CLI. This bucket is used as a temporary storage location by the Dataflow pipeline.

Use the [gsutil mb](https://cloud.google.com/storage/docs/gsutil/commands/mb) command:

gsutil mb -l REGION gs://BUCKET\_NAME

### Create a Bigtable instance and table

Create a Bigtable instance and table. For more information about creating Bigtable instances, see [Create an instance](https://cloud.google.com/bigtable/docs/creating-instance).

1. If needed, run the following command to install the [cbt CLI](https://cloud.google.com/bigtable/docs/cbt-reference):

gcloud components install cbt

1. Use the [bigtable instances create command](https://cloud.google.com/sdk/gcloud/reference/bigtable/instances/create) to create an instance:

gcloud bigtable instances create aggregate-tables \  
    --display-name=aggregate-tables \  
    --cluster-config=id=aggregate-tables-c1,zone=CLUSTER\_ZONE

**data preprocessing**

## What is data preprocessing?

Data preprocessing, a component of [data preparation](https://searchbusinessanalytics.techtarget.com/definition/data-preparation), describes any type of processing performed on [raw data](https://searchdatamanagement.techtarget.com/definition/raw-data) to prepare it for another data processing procedure. It has traditionally been an important preliminary step for the [data mining](https://searchbusinessanalytics.techtarget.com/definition/data-mining) process. More recently, data preprocessing techniques have been adapted for training machine learning models and AI models and for running inferences against them.

Data preprocessing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks. The techniques are generally used at the earliest stages of the [machine learning](https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML) and AI development pipeline to ensure accurate results.

There are several different tools and methods used for preprocessing data, including the following:

* sampling, which selects a representative subset from a large population of data;
* transformation, which manipulates raw data to produce a single input;
* denoising, which removes [noise](https://whatis.techtarget.com/definition/noise) from data;
* imputation, which synthesizes statistically relevant data for missing values;
* [normalization](https://searchsqlserver.techtarget.com/definition/normalization), which organizes data for more efficient access; and
* feature extraction, which pulls out a relevant feature subset that is significant in a particular context.

These tools and methods can be used on a variety of data sources, including data stored in files or databases and streaming data.

## Why is data preprocessing important?

Virtually any type of data analysis, [data science](https://www.techtarget.com/searchenterpriseai/definition/data-science) or AI development requires some type of data preprocessing to provide reliable, precise and robust results for enterprise applications.

Real-world data is messy and is often created, processed and stored by a variety of humans, business processes and applications. As a result, a data set may be missing individual fields, contain manual input errors, or have duplicate data or different names to describe the same thing. Humans can often identify and rectify these problems in the data they use in the line of business, but [data used to train machine learning](https://searchbusinessanalytics.techtarget.com/feature/Data-preparation-in-machine-learning-6-key-steps) or deep learning algorithms needs to be automatically preprocessed.

Machine learning and [deep learning](https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network) algorithms work best when data is presented in a format that highlights the relevant aspects required to solve a problem. Feature engineering practices that involve data wrangling, [data transformation](https://searchdatamanagement.techtarget.com/definition/data-transformation), data reduction, feature selection and feature scaling help restructure raw data into a form suited for particular types of algorithms. This can significantly reduce the processing power and time required to train a new machine learning or AI algorithm or run an inference against it.

One caution that should be observed in preprocessing data: the potential for reencoding bias into the data set. Identifying and correcting bias is critical for applications that help make decisions that affect people, such as loan approvals. Although [data scientists](https://www.techtarget.com/searchenterpriseai/definition/data-scientist) may deliberately ignore variables like gender, race or religion, these traits may be correlated with other variables like zip codes or schools attended, generating biased results.

Most modern data science packages and services now include various preprocessing libraries that help to automate many of these tasks.

## What are the key steps in data preprocessing?

The steps used in data preprocessing include the following:

**1. Data profiling**. [Data profiling](https://searchdatamanagement.techtarget.com/definition/data-profiling) is the process of examining, analyzing and reviewing data to collect statistics about its quality. It starts with a survey of existing data and its characteristics. Data scientists identify data sets that are pertinent to the problem at hand, inventory its significant attributes, and form a hypothesis of features that might be relevant for the proposed analytics or machine learning task. They also relate data sources to the relevant business concepts and consider which preprocessing libraries could be used.

**2. Data cleansing**. The aim here is to find the easiest way to rectify quality issues, such as eliminating bad data, filling in missing data or otherwise ensuring the raw data is suitable for feature engineering.

**3. Data reduction.**Raw data sets often include redundant data that arise from characterizing phenomena in different ways or data that is not relevant to a particular ML, AI or analytics task. Data reduction uses techniques like principal component analysis to transform the raw data into a simpler form suitable for particular use cases.

**4. Data transformation**. Here, data scientists think about how different aspects of the data need to be organized to make the most sense for the goal. This could include things like structuring [unstructured data](https://searchbusinessanalytics.techtarget.com/definition/unstructured-data), combining salient variables when it makes sense or identifying important ranges to focus on.

**5. Data enrichment**. In this step, data scientists apply the various feature engineering libraries to the data to effect the desired transformations. The result should be a data set organized to achieve the optimal balance between the training time for a new model and the required compute.

**6. Data validation**. At this stage, the data is split into two sets. The first set is used to train a machine learning or deep learning model. The second set is the testing data that is used to gauge the accuracy and robustness of the resulting model. This second step helps identify any problems in the [hypothesis](https://whatis.techtarget.com/definition/hypothesis) used in the cleaning and feature engineering of the data. If the data scientists are satisfied with the results, they can push the preprocessing task to a [data engineer](https://searchdatamanagement.techtarget.com/definition/data-engineer) who figures out how to scale it for production. If not, the data scientists can go back and make changes to the way they implemented the data cleansing and feature engineering steps.

## Data preprocessing techniques

There are two main categories of preprocessing -- [data cleansing](https://searchdatamanagement.techtarget.com/definition/data-scrubbing) and feature engineering. Each includes a variety of techniques, as detailed below.

## Data cleansing

Techniques for cleaning up messy data include the following:

**Identify and sort out missing data.**There are a variety of reasons a data set might be missing individual fields of data. Data scientists need to decide whether it is better to discard records with missing fields, ignore them or fill them in with a probable value. For example, in an [IoT](https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT) application that records temperature, adding in a missing average temperature between the previous and subsequent record might be a safe fix.

**Reduce**[**noisy data**](https://searchbusinessanalytics.techtarget.com/definition/noisy-data)**.** Real-world data is often noisy, which can distort an analytic or AI model. For example, a temperature sensor that consistently reported a temperature of 75 degrees Fahrenheit might erroneously report a temperature as 250 degrees. A variety of statistical approaches can be used to reduce the noise, including binning, regression and clustering.

**Identify and remove duplicates.** When two records seem to repeat, an algorithm needs to determine if the same measurement was recorded twice, or the records represent different events. In some cases, there may be slight differences in a record because one field was recorded incorrectly. In other cases, records that seem to be duplicates might indeed be different, as in a father and son with the same name who are living in the same house but should be represented as separate individuals. Techniques for identifying and removing or joining duplicates can help to automatically address these types of problems.

## Feature engineering

Feature engineering, as noted, involves techniques used by data scientists to organize the data in ways that make it more efficient to train [data models](https://searchdatamanagement.techtarget.com/definition/data-modeling) and run inferences against them. These techniques include the following:

**Feature scaling or normalization.**Often, multiple variables change over different scales, or one will change linearly while another will change exponentially. For example, salary might be measured in thousands of dollars, while age is represented in double digits. Scaling helps to transform the data in a way that makes it easier for algorithms to tease apart a meaningful relationship between variables.

**Data reduction.**Data scientists often need to combine a variety of [data sources to create a new AI or analytics model](https://searchbusinessanalytics.techtarget.com/feature/6-data-preparation-best-practices-for-analytics-applications). Some of the variables may not be correlated with a given outcome and can be safely discarded. Other variables might be relevant, but only in terms of relationship -- such as the ratio of debt to credit in the case of a model predicting the likelihood of a loan repayment; they may be combined into a single variable. Techniques like principal component analysis play a key role in reducing the number of dimensions in the training data set into a more efficient representation.

**Discretization.**It's often useful to lump raw numbers into discrete intervals. For example, income might be broken into five ranges that are representative of people who typically apply for a given type of loan. This can reduce the overhead of training a model or running inferences against it.

**Feature encoding**. Another aspect of feature engineering involves organizing unstructured data into a structured format. Unstructured data formats can include text, audio and video. For example, the process of developing natural language processing algorithms typically starts by using data transformation algorithms like Word2vec to translate words into numerical vectors. This makes it easy to represent to the algorithm that words like "mail" and "parcel" are similar, while a word like "house" is completely different. Similarly, a [facial recognition](https://www.techtarget.com/searchenterpriseai/definition/facial-recognition) algorithm might reencode raw pixel data into vectors representing the distances between parts of the face.

## How is data preprocessing used?

Data preprocessing plays a key role in earlier stages of machine learning and AI application development, as noted earlier. In an [AI](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence) context, data preprocessing is used to improve the way data is cleansed, transformed and structured to improve the accuracy of a new model, while reducing the amount of compute required.

A good data preprocessing pipeline can create reusable components that make it easier to test out various ideas for streamlining business processes or improving customer satisfaction. For example, preprocessing can improve the way data is organized for a recommendation engine by improving the age ranges used for categorizing customers.

Preprocessing can also simplify the work of creating and modifying data for more accurate and targeted business intelligence insights. For example, customers of different sizes, categories or regions may exhibit different behaviors across regions. Preprocessing the data into the appropriate forms could help BI teams weave these insights into BI dashboards.

In a customer relationship management ([CRM](https://searchcustomerexperience.techtarget.com/definition/CRM-customer-relationship-management)) context, data preprocessing is a component of web mining. Web usage logs may be preprocessed to extract meaningful sets of data called user transactions, which consist of groups of URL references. User sessions may be tracked to identify the user, the websites requested and their order, and the length of time spent on each one. Once these have been pulled out of the raw data, they yield more useful information that can be applied, for example, to consumer research, marketing or [personalization](https://searchcustomerexperience.techtarget.com/definition/personalization).

This was last updated in January 2022

#### Continue Reading About data preprocessing

* [Top data preparation challenges and how to overcome them](https://searchbusinessanalytics.techtarget.com/feature/Top-data-preparation-challenges-and-how-to-overcome-them)
* [Big data teams get proactive on data preparation for analytics users](https://searchdatamanagement.techtarget.com/feature/Big-data-teams-get-proactive-on-data-preparation-for-analytics-users)
* [IoT data lifecycle adapts to AI at the edge](https://internetofthingsagenda.techtarget.com/feature/IoT-data-lifecycle-adapts-to-AI-at-the-edge)
* [Deft preprocessing marks deep learning techniques for data preparation](https://searchdatamanagement.techtarget.com/podcast/Deft-preprocessing-marks-deep-learning-techniques-for-data-preparation)
* [Self-service data preparation: What it is and how it helps users](https://searchdatamanagement.techtarget.com/feature/Self-service-data-preparation-What-it-is-and-how-it-helps-users?_gl=1*9lgdsl*_ga*NTA0MjU2MDcuMTY5NzE3ODc3OA..*_ga_TQKE4GS5P9*MTY5NzE3ODc3Ny4xLjEuMTY5NzE3OTA1NS4wLjAuMA..&_ga=2.38248545.376883778.1697178779-50425607.1697178778)