



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** X **Month of publication:** October 2025

DOI: <https://doi.org/10.22214/ijraset.2025.74554>

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

DataOps in Data Analysis world

Justin Pathrose James

LA Care Health Plan, USA

DataOps is widely used in data-driven decision-making sectors, such as finance, healthcare, and e-commerce. It supports use cases like real-time analytics, machine learning model deployment, and regulatory compliances.

I. DATA ANALYSIS

Why it is required?

In daily life, there are numerous things that require addition to them. In order to make additions or enhancements, the market must be analyzed, and a proper idea should be gained regarding what is really required. If a product is launched without knowing the market properly, it will be useless. Data analysis is the best method of analyzing a market and its actual requirements.

II. CASE STUDY

A new company wants to introduce a lactose free product in the city.

They are trying to find out the information grocery stores provide.

- 1) Data points collected to identify target audience
 - Average number of transactions conducted for lactose free products.
 - Age group of consumers
 - Number of cases of lactose intolerance reported to health care centers.
 - Age group of patients
- 2) Feasibility study
 - Leading product
 - Survey for families looking for new product.
 - Complaints filed for current leading product
- 3) Cost Analysis
 - Current leading product price

III. HOW CAN DATA OPS BE USED FOR THIS SURVEY ?

A. Data Ops

Data Operations (DataOps) is a set of practices and methodologies for increasing the efficiency, quality, and collaboration of data managing and analyzing. It draws inspiration from DevOps and Agile methodologies and is focused on automating and streamlining data pipelines to deliver consistent, high-quality data for business intelligence and analytics.

B. DataOps in Data Analysis world

DataOps encompasses the entire data lifecycle, from data ingestion and transformation to analysis and reporting. DataOps is all about breaking down silos between data producers (e.g., engineers) and consumers (e.g., analysts) to facilitate collaboration and seamless data flow.

This example uses Python for data processing, Great Expectations for data quality, dbt for data modeling, and Apache Airflow for workflow orchestration.

1) Raw survey data

Assume a survey tool exports raw responses into a CSV file, `lactose_survey_raw.csv`.

csv

timestamp	participant_id	gender	age_group	lactose_intolerant_diagnosis	dairy_consumption_freq	lactose_free_usage
2025-08-01 09:15:21	P001	Male	25-34	Yes	Daily	Sometimes
2025-08-01 09:17:45	P002	Female	35-44	No	Once a week	Never
2025-08-01 09:19:02	P003	Female	18-24	No	Several times a week	Never
2025-08-01 09:21:55	P004	Other	45-54	Maybe	Daily	Always
2025-08-01 09:23:10	P005	Male	18-24	Yes	Never	Always

2) Python script for data processing (ingestion and cleaning)

This script ingests the CSV, performs basic cleaning like standardizing lactose_intolerant_diagnosis, and renames columns. This task is run by Airflow.

Python

```
import pandas as pa
def process_data(input_path, output_path):
    """Ingests raw survey data, cleans it, and saves the cleaned version."""
    df = pa.read_csv(input_path)

    # Standardize 'lactose_intolerant_diagnosis' column
    df['lactose_intolerant_diagnosis'] = df['lactose_intolerant_diagnosis'].replace(
        {'Yes': True, 'No': False, 'Maybe': None}
    )

    # Convert timestamp to datetime and extract date
    df['timestamp'] = pa.to_datetime(df['timestamp'])
    df['response_date'] = df['timestamp'].dt.date

    # Rename columns for clarity
    df.rename(columns={
        'dairy_consumption_freq': 'dairy_consumption_frequency',
        'lactose_free_usage': 'lactose_free_usage_frequency'
    }, inplace=True)

    # Save cleaned data to an interim file
    df.to_csv(output_path, index=False)
    print(f"Cleaned data saved to {output_path}")

if __name__ == "__main__":
    import os
    # Example paths - in a real scenario, these would be managed by Airflow
    RAW_DATA_PATH = 'data/lactose_survey_raw.csv'
    CLEANED_DATA_PATH = 'data/lactose_survey_cleaned.csv'
```

```
# Ensure data directory exists
```

```
os.makedirs(os.path.dirname(CLEANED_DATA_PATH), exist_ok=True)
```

```
process_data(RAW_DATA_PATH, CLEANED_DATA_PATH)
```

3) Data quality check with Great Expectations

This Python script validates the cleaned data to ensure it meets quality standards before transformation.

Python

```
import great_expectations as gx
```

```
import os
```

```
import sys
```

```
def validate_data(data_path):
```

```
    """Validates the cleaned survey data using Great Expectations."""
```

```
    context = gx.get_context()
```

```
# Load the data
```

```
df = pa.read_csv(data_path)
```

```
# Create a GX validator
```

```
validator = context.sources.pandas_default.read_dataframe(df)
```

```
# Define expectations
```

```
validator.expect_column_to_exist(column="participant_id")
```

```
validator.expect_column_values_to_be_unique(column="participant_id")
```

```
validator.expect_column_values_to_not_be_null(column="participant_id")
```

```
validator.expect_column_values_to_be_in_set(
```

```
    column="gender",
```

```
    value_set=['Male', 'Female', 'Other']
```

```
)
```

```
validator.expect_column_values_to_be_in_set(
```

```
    column="lactose_free_usage_frequency",
```

```
    value_set=['Always', 'Sometimes', 'Never']
```

```
)
```

```
# The 'diagnosis' column can be null for 'Maybe' answers
```

```
validator.expect_column_values_to_be_in_set(
```

```
    column="lactose_intolerant_diagnosis",
```

```
    value_set=[True, False],
```

```
    mostly=0.9
```

```
)
```

```
# Run validation and save the report
```

```
result = validator.validate(
```

```
    result_format="SUMMARY",
```

```
    only_return_failures=False
```

```
)
```

```
if not result["success"]:
```

```
    print("Data quality check failed!")
```

```
    print(result)
```



```
sys.exit(1)
else:
    print("Data quality check passed successfully!")

if __name__ == "__main__":
    CLEANED_DATA_PATH = 'data/lactose_survey_cleaned.csv'
    validate_data(CLEANED_DATA_PATH)
```

4) Data modeling with dbt

This SQL model transforms the cleaned survey data into an aggregated model ready for downstream analytics.

SQL

```
{{ config(materialized='table') }}
```



```
WITH base_survey AS (
SELECT
participant_id,
gender,
age_group,
lactose_intolerant_diagnosis,
dairy_consumption_frequency,
lactose_free_usage_frequency
FROM
{{ ref('lactose_survey_cleaned') }} -- References the table from the cleaned CSV
)

SELECT
age_group,
gender,
lactose_intolerant_diagnosis,
dairy_consumption_frequency,
lactose_free_usage_frequency,
count(participant_id) as participant_count
FROM
base_survey
GROUP BY
1, 2, 3, 4, 5
ORDER BY
age_group, gender
```

5) Orchestration with Apache Airflow DAG

This Airflow DAG defines the entire DataOps pipeline, ensuring each step is run in the correct order and dependencies are met.

Python

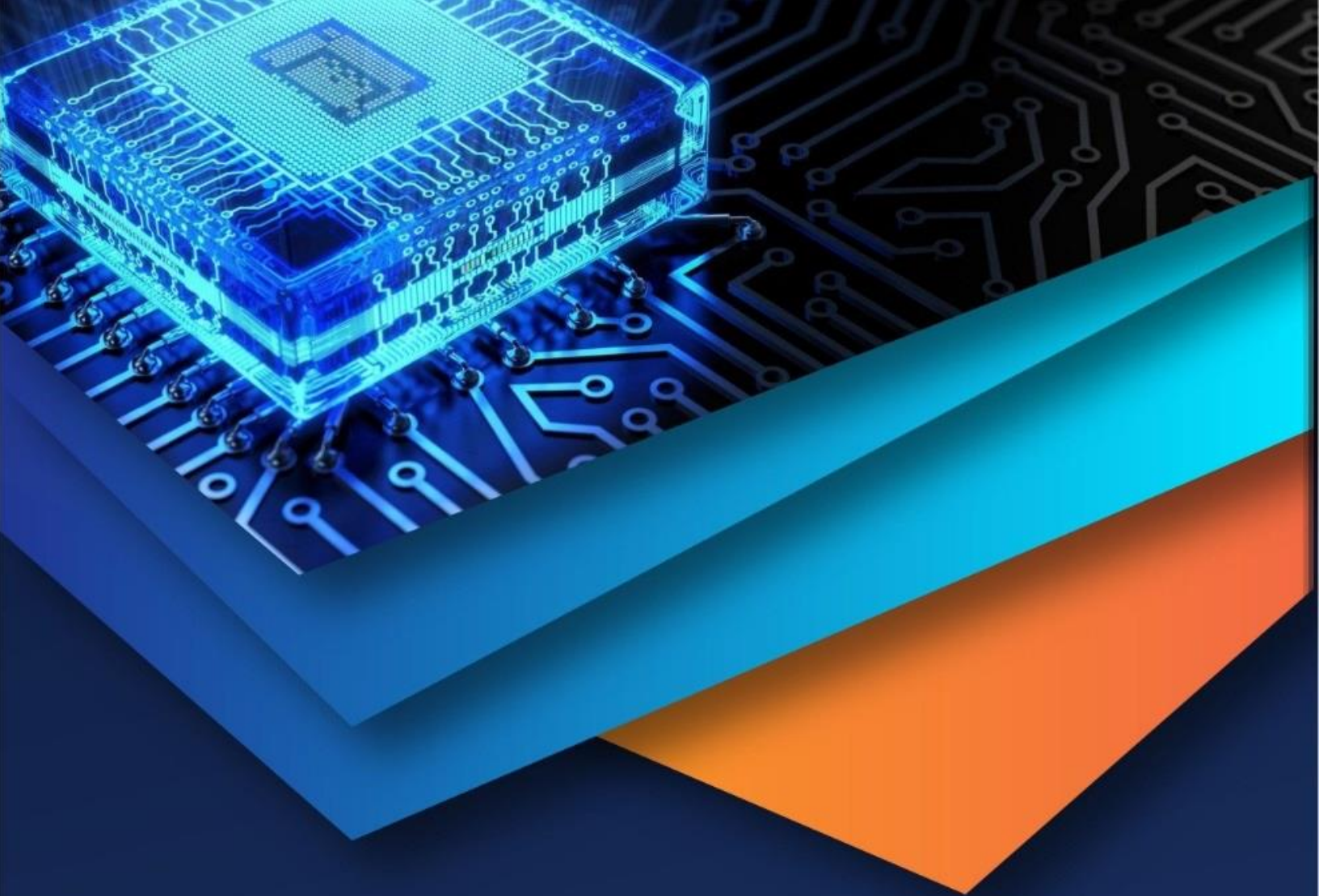
```
from __future__ import annotations
import pendulum
from airflow.models.dag import DAG
from airflow.operators.python import PythonOperator
from airflow.operators.bash import BashOperator

with DAG(
    dag_id="lactose_survey_pipeline",
```



```
start_date=pendulum.datetime(2025, 8, 1, tz="UTC"),
catchup=False,
tags=["dataops", "survey", "lactose"],
schedule=None, # Run manually for this example
) as dag:
# 1. Task to process and clean the raw survey data
process_raw_data = PythonOperator(
    task_id="process_raw_data",
    python_callable=process_data,
    op_kwargs={
        'input_path': '/opt/airflow/data/lactose_survey_raw.csv',
        'output_path': '/opt/airflow/data/lactose_survey_cleaned.csv'
    },
)

# 2. Task to validate the cleaned data
validate_cleaned_data = PythonOperator(
    task_id="validate_cleaned_data",
    python_callable=validate_data,
    op_kwargs={
        'data_path': '/opt/airflow/data/lactose_survey_cleaned.csv'
    },
)
```

10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)