UDEMY COURSE ANALYSIS REPORT

- Shubhang Yadav Sandaveni

Introduction

Online learning platforms have revolutionized education by providing accessibility and flexibility for learners worldwide. Udemy, as a leading online course provider, offers thousands of courses on diverse topics ranging from business and technology to lifestyle and arts. However, the abundance of data generated by the platform presents challenges in deriving meaningful insights to improve course offerings, pricing strategies, and overall user satisfaction. This project leverages data warehousing and ETL (Extract, Transform, Load) processes using AWS services to design and implement a robust data analytics solution. The result is an automated, scalable, and actionable platform that provides in-depth insights into course performance, user feedback, and instructor contributions.

Problem Statement

Udemy's vast repository of course data is an invaluable resource for driving business decisions, but it lacks a centralized, structured, and automated data pipeline for analysis. Key challenges include:

1. Unstructured and raw data formats from various sources.
2. Lack of an efficient querying mechanism for extracting actionable insights.
3. Manual data processing, which is prone to errors and inefficiencies.
4. Limited visibility into key metrics such as course performance, user feedback, and revenue generation.

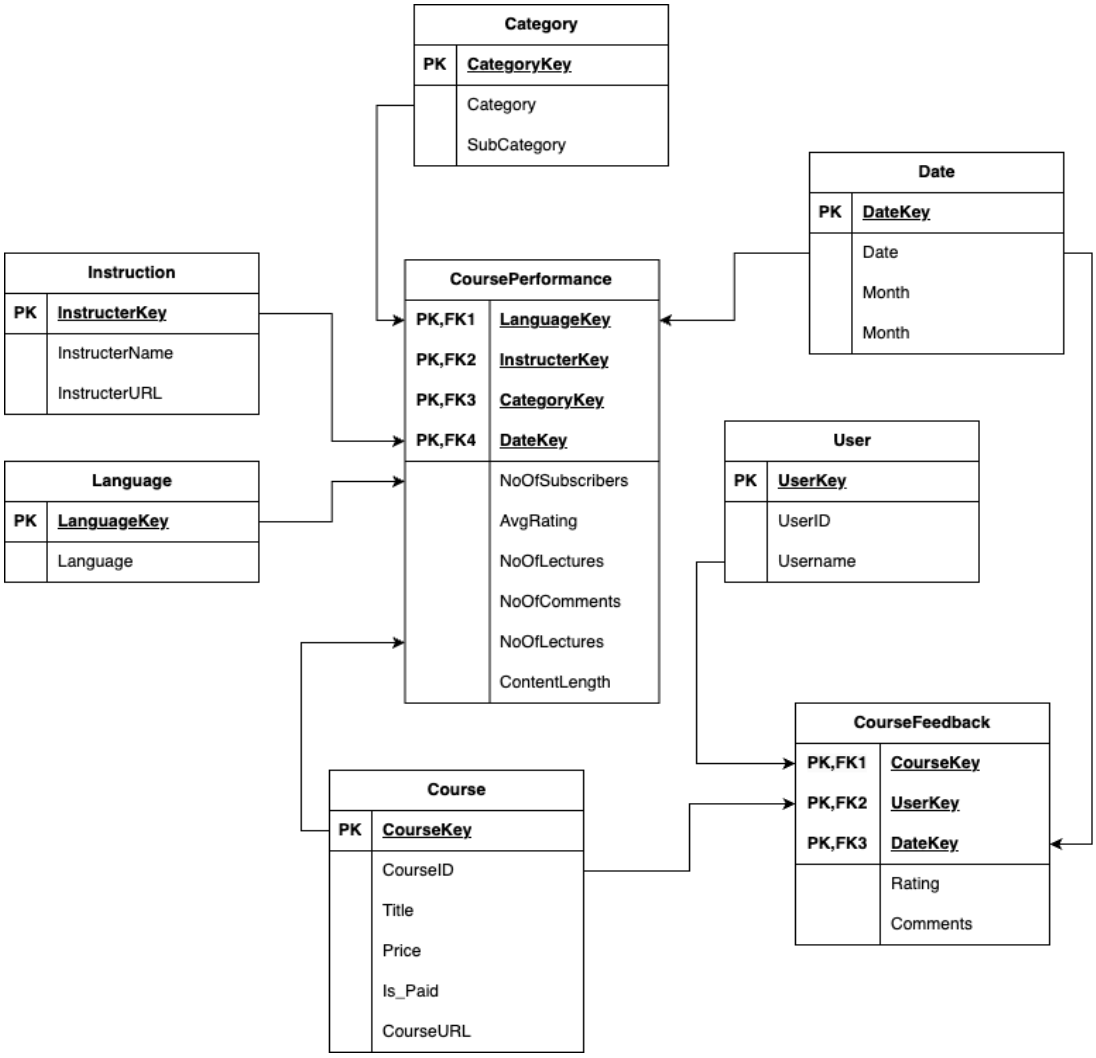
These issues hinder the platform's ability to evaluate course quality, optimize pricing, and support instructors in improving their content. A scalable data warehouse and visualization system are required to address these challenges.

Project Objectives

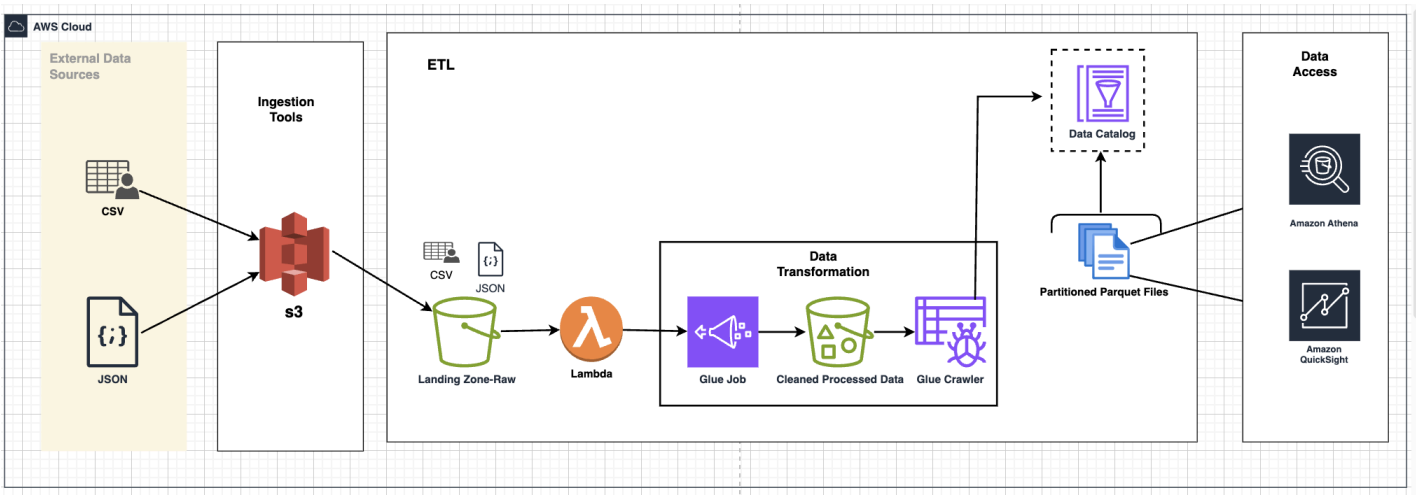
The primary objective of this project is to create an automated data pipeline and a data warehouse for Udemy course analysis. Specific goals include:

1. **Data Centralization**:
    - Collect and store raw datasets (comments, courses, and metadata) in a scalable and secure location using AWS S3.
2. **Data Transformation**:
    - Design ETL processes to clean, transform, and structure raw data into well-defined fact and dimension tables for analysis.
3. **Schema Design**:
    - Develop a robust multi-dimensional schema to facilitate OLAP operations and comprehensive analysis.
4. **Automation**:
    - Automate data ingestion, transformation, and loading processes using AWS Glue and AWS Lambda to ensure real-time updates and minimal manual intervention.
5. **Analysis and Visualization**:
    - Use AWS Athena for querying and Tableau for creating interactive visualizations to provide actionable insights into course performance, user feedback, and revenue trends.
6. **Key Metrics and Insights**:
    - Develop KPIs to evaluate course pricing, instructor contributions, user satisfaction, and content structure, enabling Udemy to make data-driven decisions.

Multi Dimension Logical Model



AWS Data Warehouse Architecture



Pipeline Flow

1. Data Ingestion

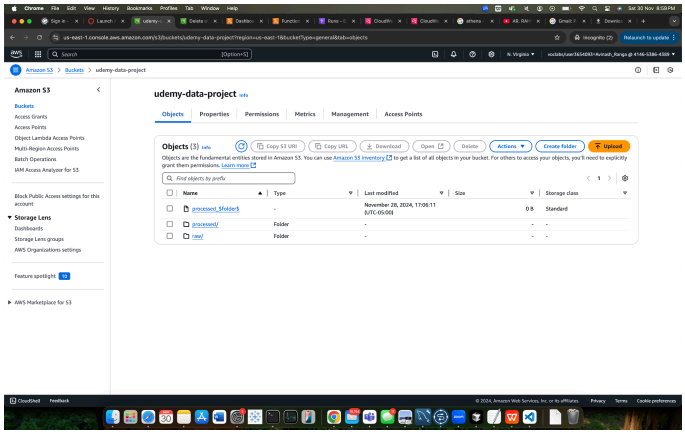
Source Files:

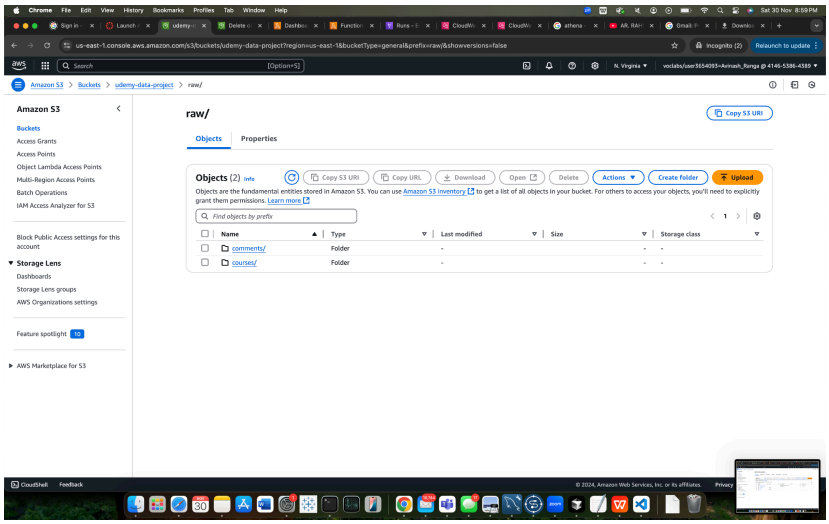
Comments.csv: Contains user feedback, including comments, ratings, and timestamps.

Courses_info.json : Contains metadata about courses such as title, price, enrollment, and category.

Storage:

Uploaded the raw datasets to an AWS S3 bucket named Udemy-data-project under the raw folder. This folder acts as the starting point of the ETL pipeline.





---

2. Data Transformation

ETL Process:

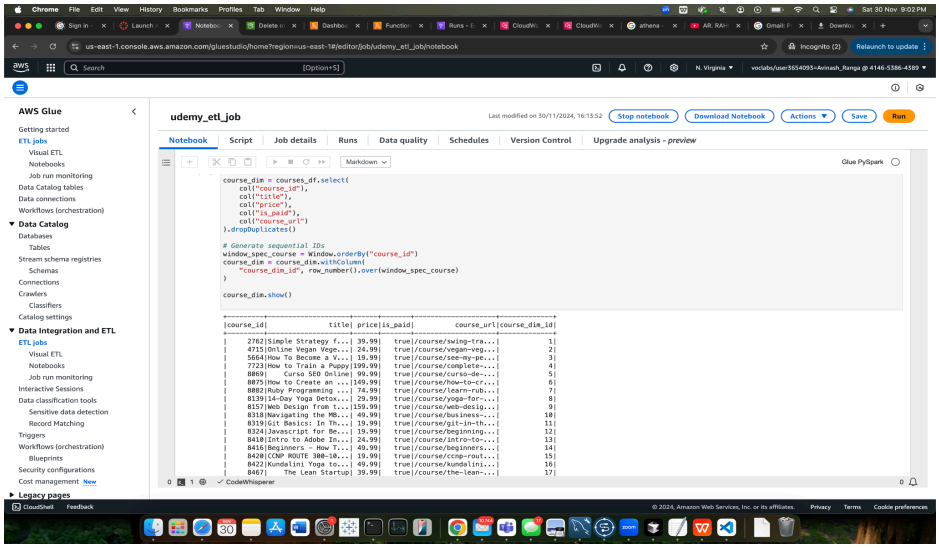
AWS Glue was used to perform the ETL (Extract, Transform, Load) operations using Python and PySpark. Key transformation steps include:

Data Cleaning:

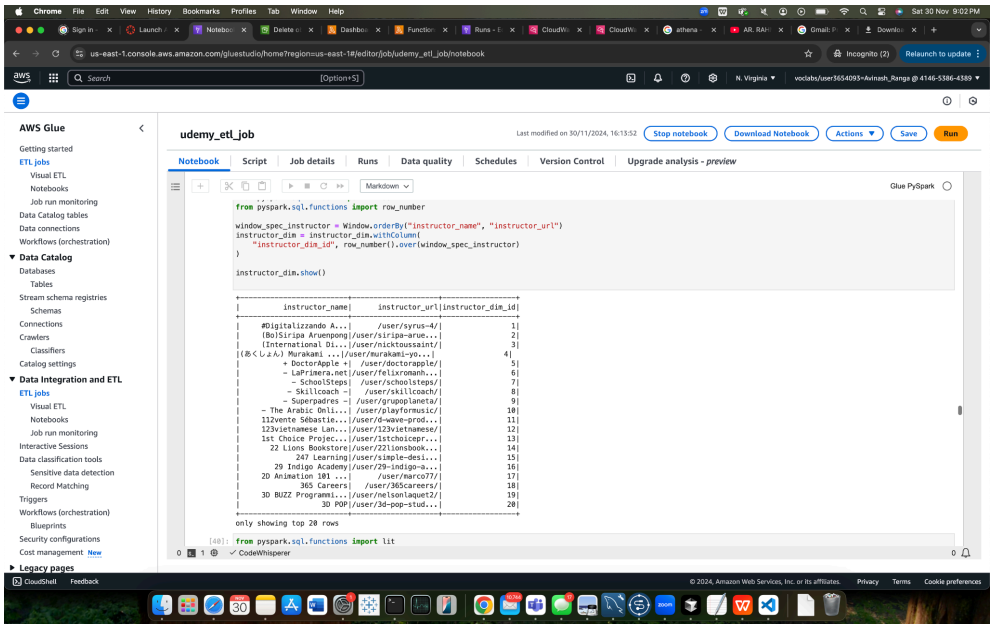
- Null values in critical columns were either removed or replaced.
- Inconsistent formats (e.g., dates) were standardized to ensure data integrity.

1.Dimension Table Creation:

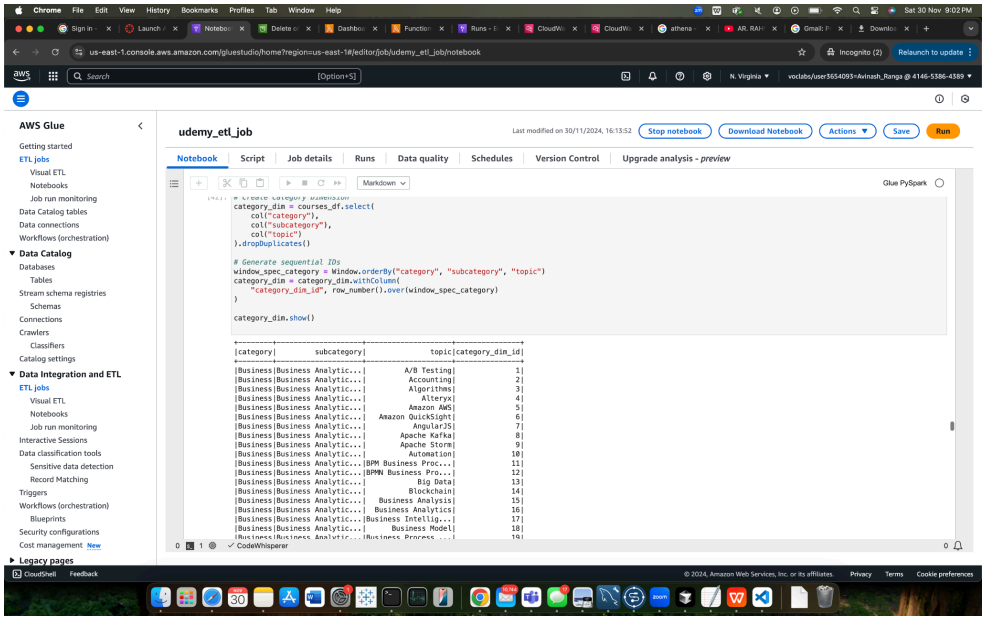
Course Dimension: Contains static course attributes like course title, price, and is_paid status.



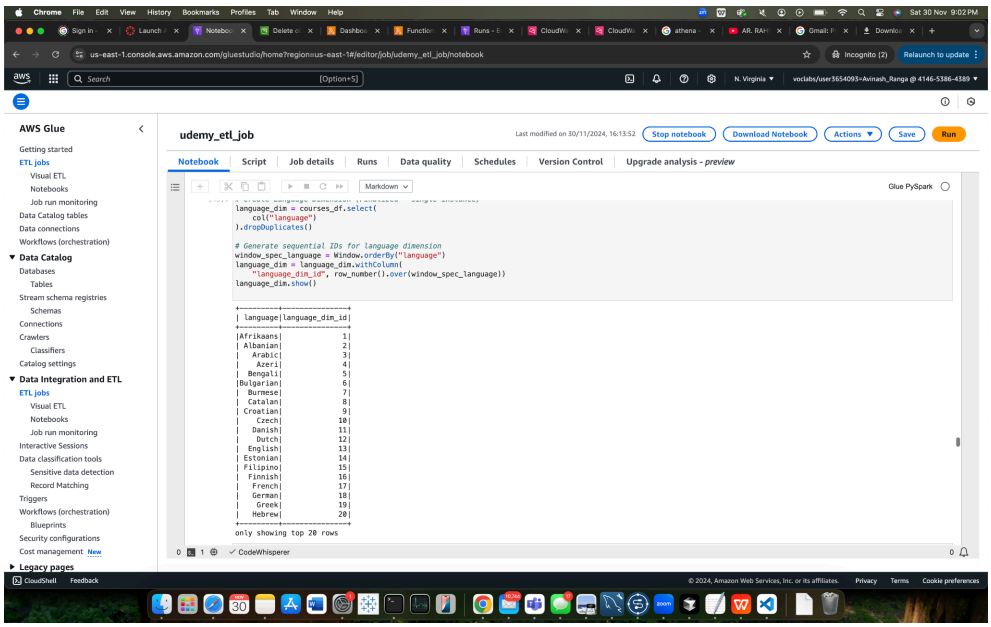
Instructor Dimension: Details about instructors.



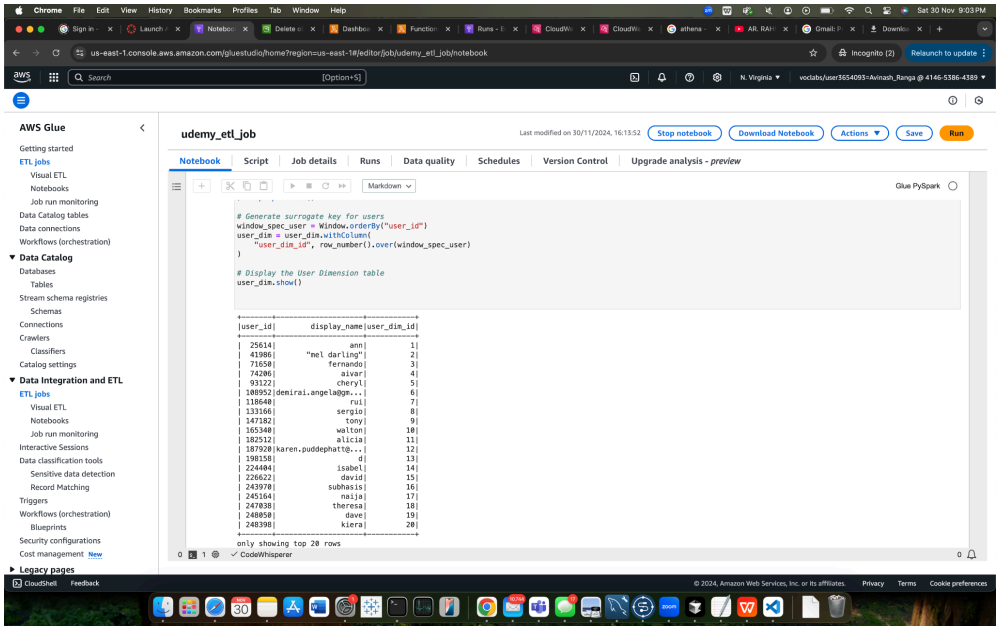
Category Dimension: Tracks hierarchical categories (category, subcategory, topic).



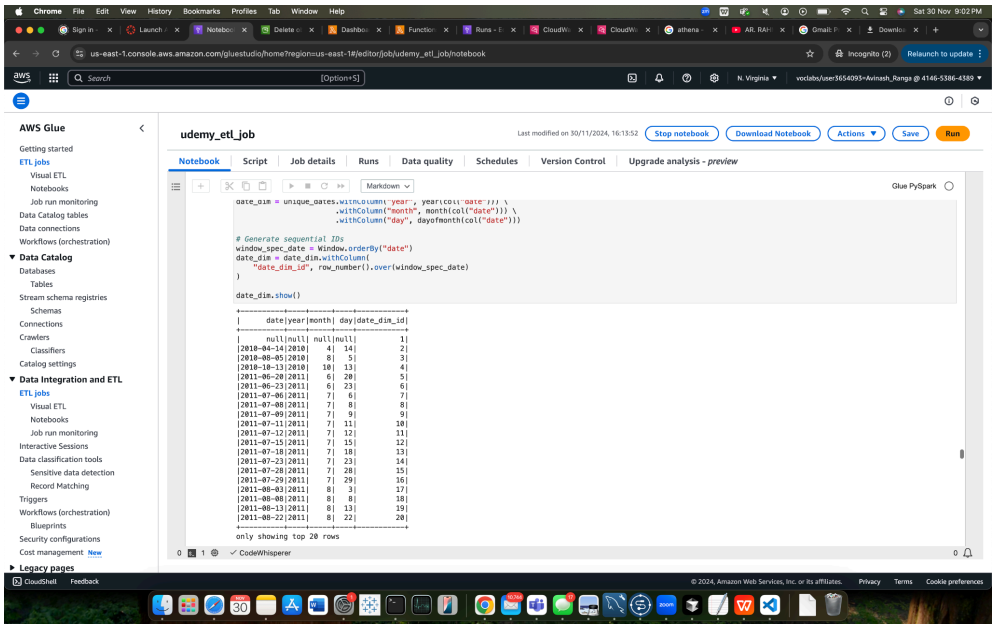
Language Dimension: Tracks the course's language.



User Dimension: Stores user-specific information like user ID and display name.

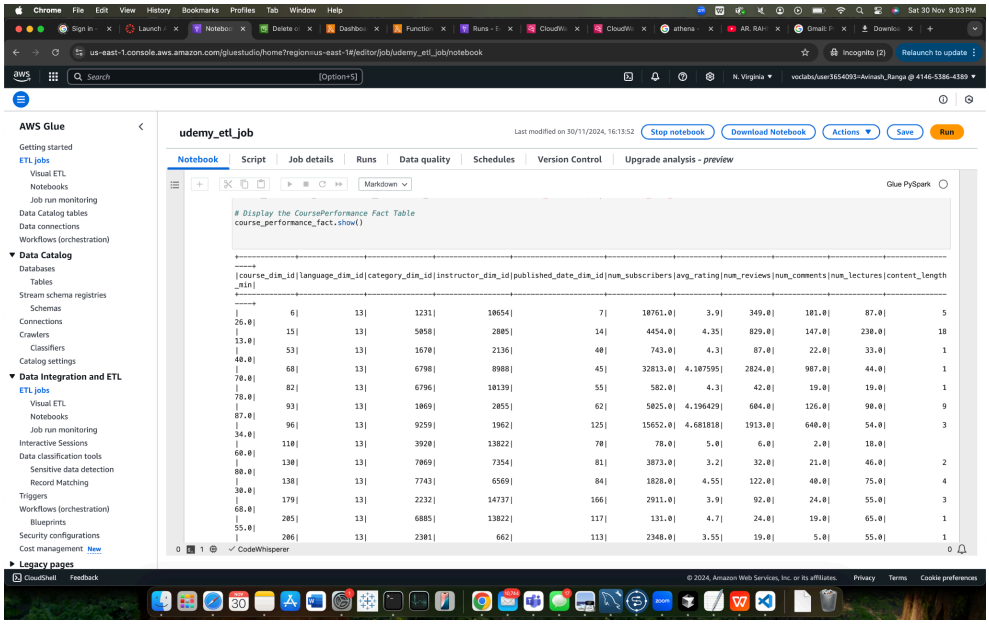


Date Dimension: Tracks dates for feedback and course-related events.

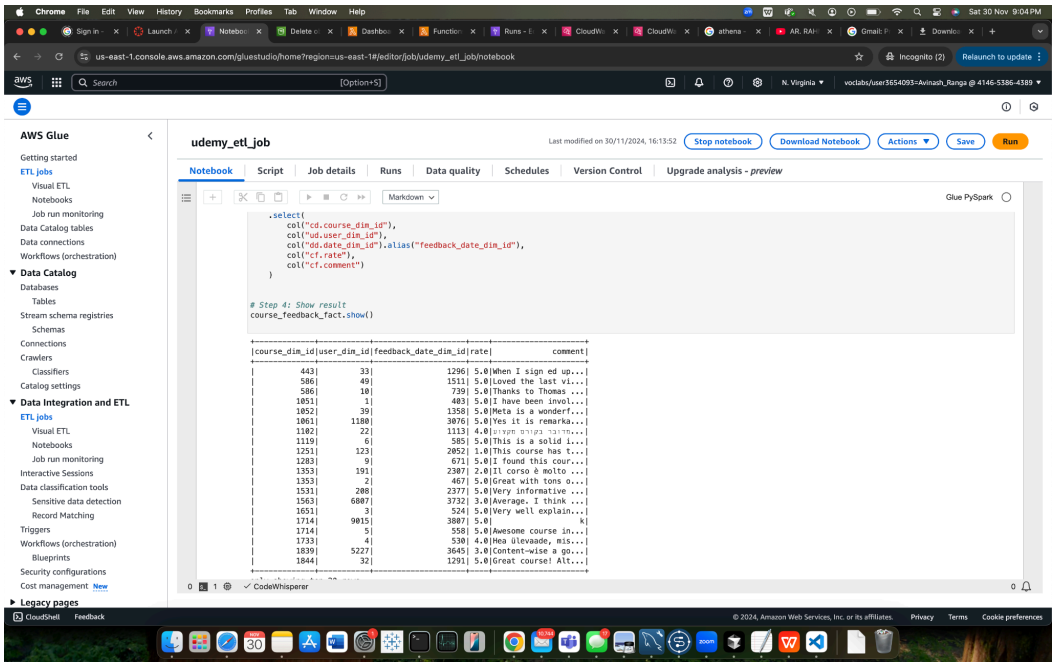


Fact Table Creation:

- **CoursePerformance Fact Table**: Aggregated metrics like total subscribers, average ratings, total reviews, and course duration in minutes.



- **<u>CourseFeedback Fact Table</u>**: Granular data capturing individual user ratings and comments for each course.



Processed Data Storage:

The transformed and structured data was stored in the processed folder in the same S3 bucket for querying and analysis.

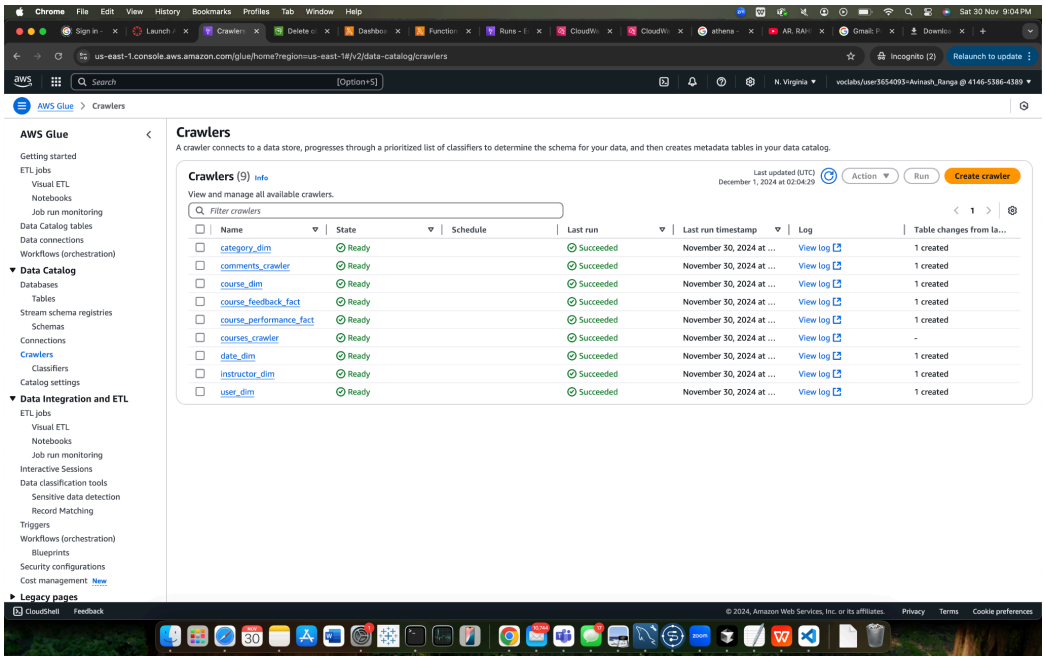
Key PySpark Operations:

- Aggregate functions like count, avg, and sum were used to calculate metrics for the fact tables.
- Schema enforcement and trimming operations ensured the data adhered to a consistent format.

3. Data Loading

Glue Crawlers:

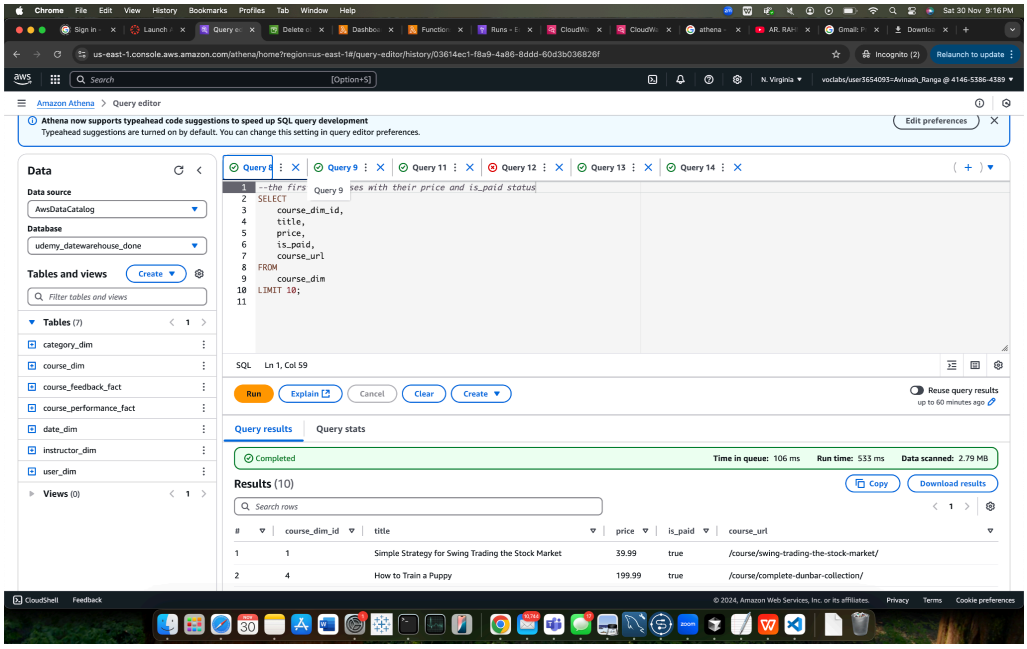
- AWS Glue Crawlers were configured to scan the processed data in the S3 bucket and generate metadata tables in the udemy_datawarehouse_etl database.
- These tables represent the schema for both fact and dimension tables, enabling seamless integration with Athena for querying.

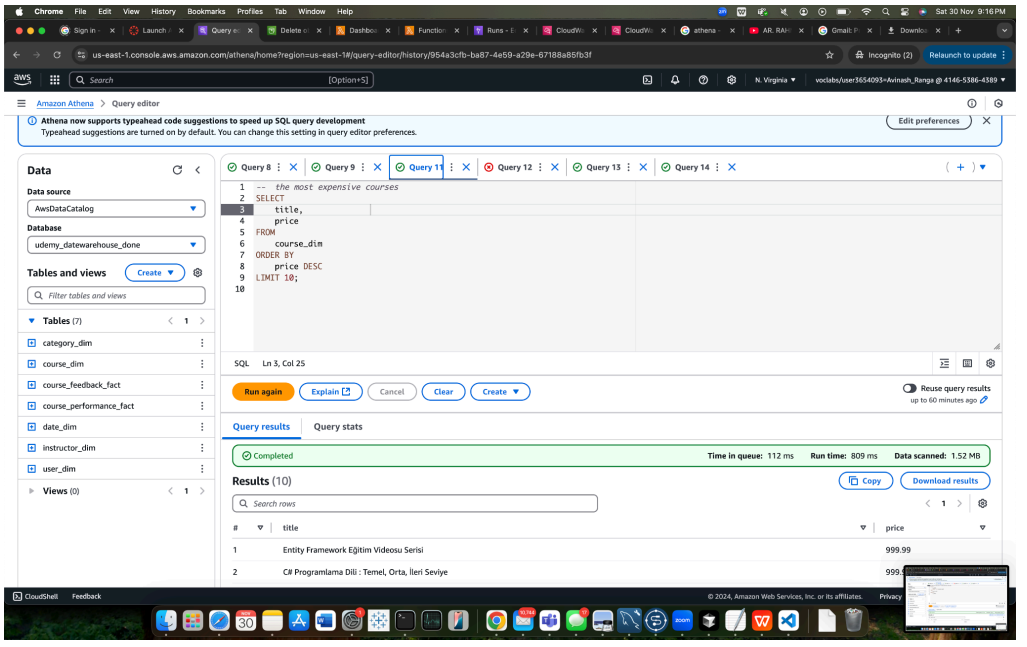


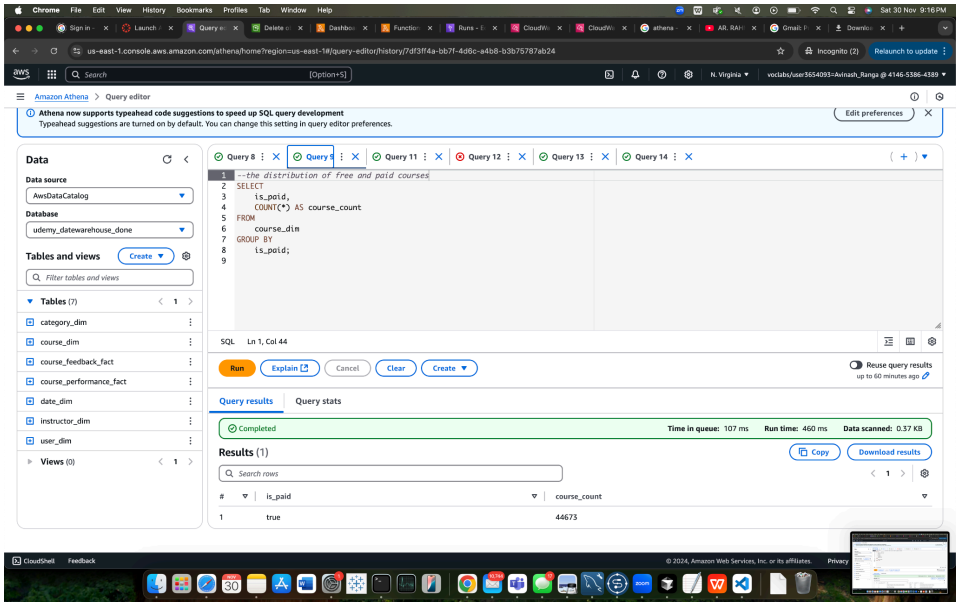
4. Querying and Analysis

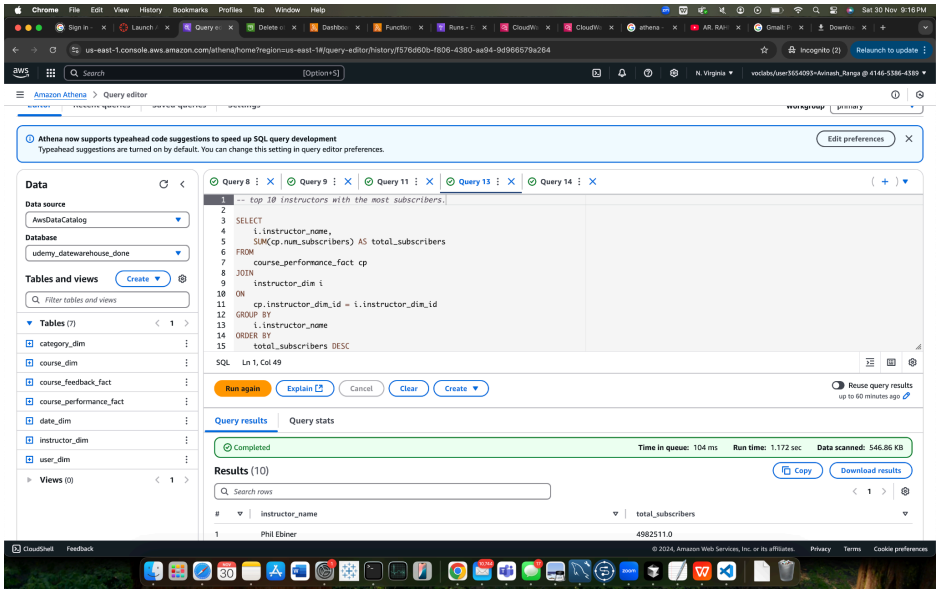
AWS Athena:

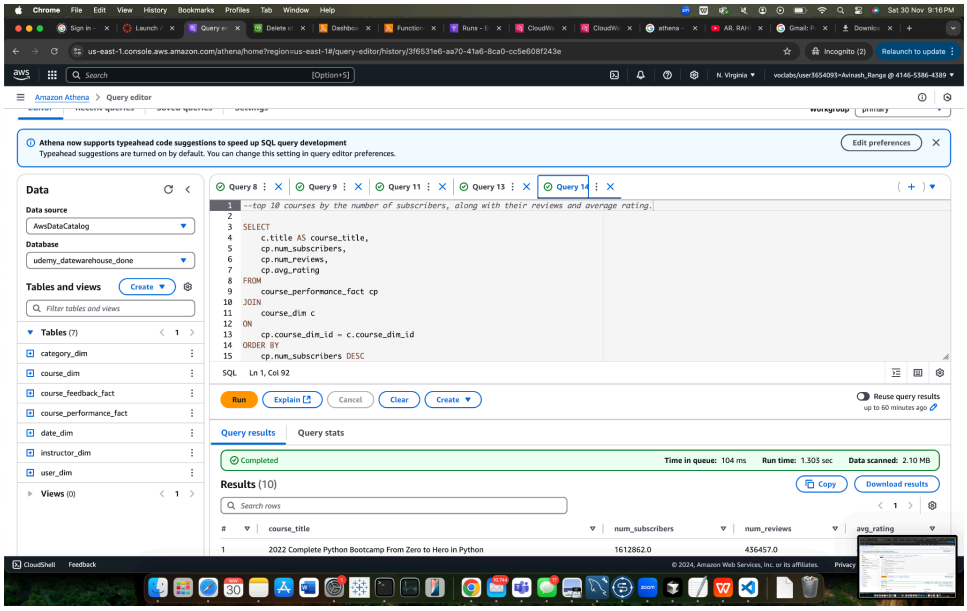
- The structured data was queried using AWS Athena to extract actionable insights. Examples of queries include:
- Listing top-performing courses based on average ratings and subscriber count.
- Analyzing trends in user feedback by category, language, and date.


Amazon Athena > Query editor
Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.
Edit preferences
Data
Data source
AwsDataCatalog
Database
udemy_datewarehouse_done
Tables and views
Create
Filter tables and views
Tables (7)
category_dim
course_dim
course_feedback_fact
course_performance_fact
date_dim
instructor_dim
user_dim
Views (0)
Query 8
Query 9
Query 11
Query 12
Query 13
Query 14
1 --the firs Query 9 ses with their price and is_paid status
2 SELECT
3 course_dim_id,
4 title,
5 price,
6 is_paid,
7 course_url
8 FROM
9 course_dim
10 LIMIT 10;
11
SQL Ln 1, Col 59
Run
Explain
Cancel
Clear
Create
Reuse query results
up to 60 minutes ago
Query results
Query stats
Completed
Time in queue: 106 ms
Run time: 533 ms
Data scanned: 2.79 MB
Results (10)
Copy
Download results
Search rows
course_dim_id title price is_paid course_url
1 1 Simple Strategy for Swing Trading the Stock Market 39.99 true /course/swing-trading-the-stock-market/
2 4 How to Train a Puppy 199.99 true /course/complete-dunbar-collection/
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Amazon Athena > Query editor
Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.
Edit preferences
Data
Data source
AwsDataCatalog
Database
udemy_datewarehouse_done
Tables and views
Create
Filter tables and views
Tables (7)
category_dim
course_dim
course_feedback_fact
course_performance_fact
date_dim
instructor_dim
user_dim
Views (0)
Query 8
Query 9
Query 11
Query 12
Query 13
Query 14
1 -- the most expensive courses
2 SELECT
3 title,
4 price
5 FROM
6 course_dim
7 ORDER BY
8 price DESC
9 LIMIT 10;
10
SQL Ln 3, Col 25
Run again
Explain
Cancel
Clear
Create
Reuse query results
up to 60 minutes ago
Query results
Query stats
Completed
Time in queue: 112 ms
Run time: 809 ms
Data scanned: 1.52 MB
Results (10)
Copy
Download results
Search rows
title price
1 Entity Framework Eğitim Videosu Serisi 999.99
2 C# Programlama Dili : Temel, Orta, İleri Seviye
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy


Amazon Athena > Query editor
Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.
Edit preferences
Data
Data source
AwsDataCatalog
Database
udemy_datewarehouse_done
Tables and views
Create
Filter tables and views
Tables (7)
category_dim
course_dim
course_feedback_fact
course_performance_fact
date_dim
instructor_dim
user_dim
Views (0)
Query 8
Query 9
Query 11
Query 12
Query 13
Query 14
1 --the distribution of free and paid courses
2 SELECT
3 is_paid,
4 COUNT(*) AS course_count
5 FROM
6 course_dim
7 GROUP BY
8 is_paid;
9
SQL Ln 1, Col 44
Run
Explain
Cancel
Clear
Create
Reuse query results
up to 60 minutes ago
Query results
Query stats
Completed
Time in queue: 107 ms
Run time: 460 ms
Data scanned: 0.37 KB
Results (1)
Copy
Download results
Search rows
is_paid course_count
1 true 44673
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy


Amazon Athena > Query editor
Athena now supports typeahead code suggestions to speed up SQL query development
Typeahead suggestions are turned on by default. You can change this setting in query editor preferences.
Edit preferences
Data
Data source
AwsDataCatalog
Database
udemy_datewarehouse_done
Tables and views
Create
Filter tables and views
Tables (7)
category_dim
course_dim
course_feedback_fact
course_performance_fact
date_dim
instructor_dim
user_dim
Views (0)
Query 8
Query 9
Query 11
Query 13
Query 14
1 -- top 10 instructors with the most subscribers.
2
3 SELECT
4 i.instructor_name,
5 SUM(cp.num_subscribers) AS total_subscribers
6 FROM
7 course_performance_fact cp
8 JOIN
9 instructor_dim i
10 ON
11 cp.instructor_dim_id = i.instructor_dim_id
12 GROUP BY
13 i.instructor_name
14 ORDER BY
15 total_subscribers DESC
SQL Ln 1, Col 49
Run again
Explain
Cancel
Clear
Create
Reuse query results
up to 60 minutes ago
Query results
Query stats
Completed
Time in queue: 104 ms
Run time: 1.172 sec
Data scanned: 546.86 KB
Results (10)
Copy
Download results
Search rows
instructor_name total_subscribers
1 Phil Ebiner 4982511.0
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



Key Benefits:

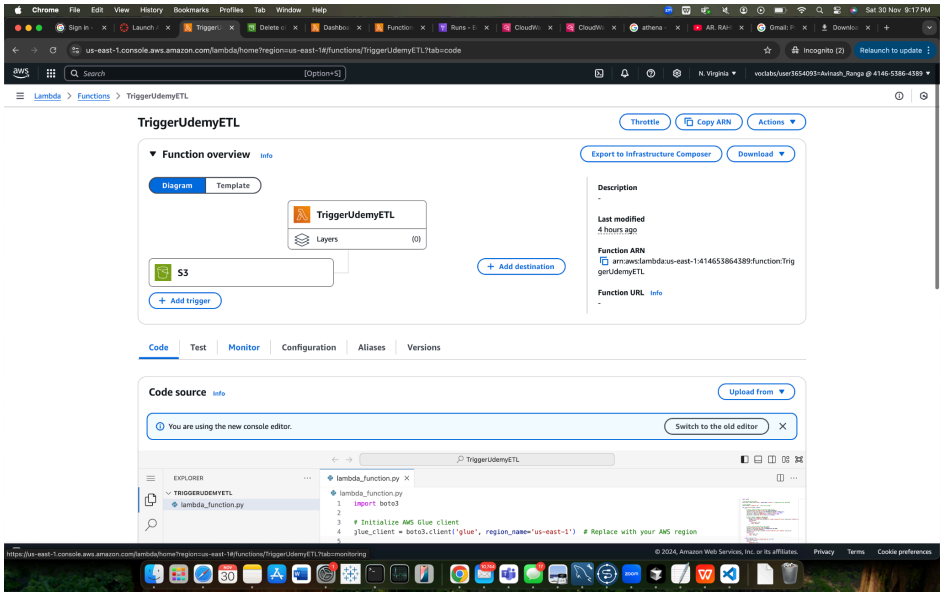
- Serverless query execution with a pay-per-query model.
- Efficient OLAP-style querying over large datasets stored in S3.

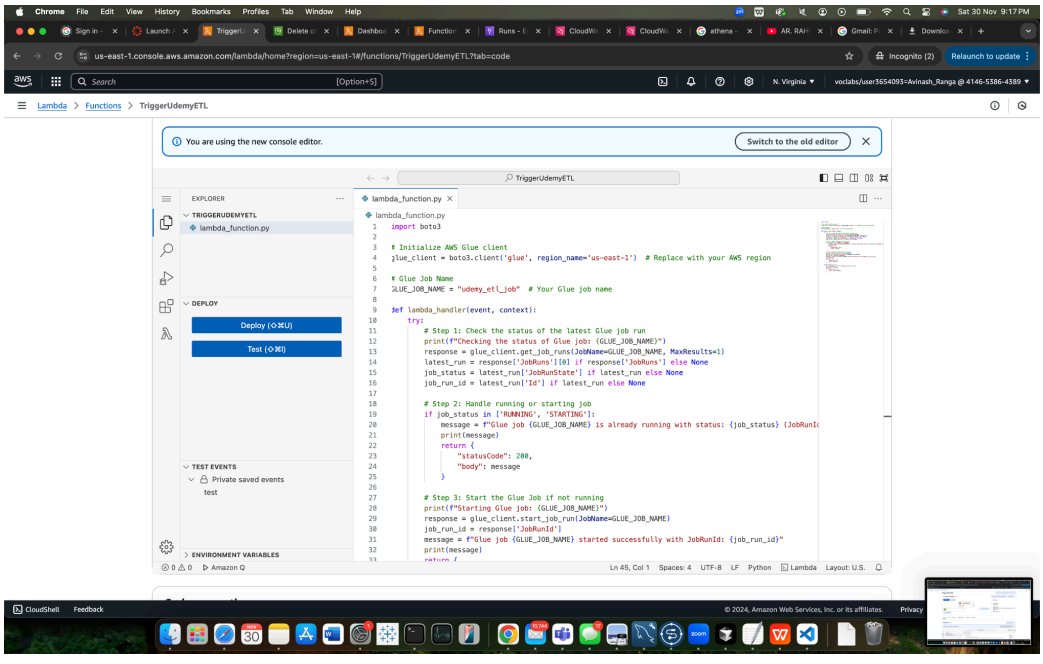
5. Pipeline Automation

AWS Lambda:

A Lambda function was created to automate the pipeline:

- **Trigger**: Monitors the S3 bucket for new file uploads in the raw folder.
- **Glue Integration**: Automatically triggers the Glue ETL job to process the new data.
- **Error Handling**: Tracks the status of Glue jobs and ensures successful execution or error resolution.


TriggerUdemyETL
Throttle
Copy ARN
Actions
Function overview
Export to Infrastructure Composer
Download
Diagram
Template
TriggerUdemyETL
Layers
S3
Add trigger
Add destination
Description
Last modified
4 hours ago
Function ARN
arn:aws:lambda:us-east-1:414653864389:function:TriggerUdemyETL
Function URL
Code
Test
Monitor
Configuration
Aliases
Versions
Code source
Upload from
You are using the new console editor.
Switch to the old editor
lambda_function.py
import boto3
# Initialize AWS Glue client
glue_client = boto3.client('glue', region_name='us-east-1') # Replace with your AWS region


You are using the new console editor.
Switch to the old editor
EXPLORER
TRIGGERUDEMYETL
lambda_function.py
DEPLOY
Deploy (⌥⌘U)
Test (⌥⌘I)
TEST EVENTS
Private saved events
test
ENVIRONMENT VARIABLES
import boto3
# Initialize AWS Glue client
glue_client = boto3.client('glue', region_name='us-east-1') # Replace with your AWS region
# Glue Job Name
GLUE_JOB_NAME = "udemy_etl_job" # Your Glue job name
def lambda_handler(event, context):
try:
# Step 1: Check the status of the latest Glue job run
print(f"Checking the status of Glue job: {GLUE_JOB_NAME}")
response = glue_client.get_job_runs(JobName=GLUE_JOB_NAME, MaxResults=1)
latest_run = response['JobRuns'][0] if response['JobRuns'] else None
job_status = latest_run['JobRunState'] if latest_run else None
job_run_id = latest_run['Id'] if latest_run else None
# Step 2: Handle running or starting job
if job_status in ['RUNNING', 'STARTING']:
message = f"Glue job {GLUE_JOB_NAME} is already running with status: {job_status}
print(message)
return {
"statusCode": 200,
"body": message
}
# Step 3: Start the Glue Job if not running
print(f"Starting Glue job: {GLUE_JOB_NAME}")
response = glue_client.start_job_run(JobName=GLUE_JOB_NAME)
job_run_id = response['JobRunId']
message = f"Glue job {GLUE_JOB_NAME} started successfully with JobRunId: {job_run_id}"
print(message)
CloudShell
Feedback

6. Schema Design

Fact Tables

1.CoursePerformance Fact Table:

Metrics: num_subscribers, avg_rating, num_reviews, num_comments, num_lectures, content_length_min.

2.CourseFeedback Fact Table:

Metrics: rate and comment.

Dimension Tables

- **Course Dimension**: Static attributes of courses such as course ID, title, price, and is_paid status.
- **Instructor Dimension**: Instructor-specific details.
- **Category Dimension**: Hierarchical categories, including category, subcategory, and topic.
- **Language Dimension**: Tracks the language of courses.
- **User Dimension**: Contains user-specific information such as user ID and display name.
- **Date Dimension**: Tracks dates associated with feedback and course events.

Pipeline Tools

1.AWS S3:

- Primary storage for raw and processed data.
- Enables centralized and scalable storage for the pipeline.

2.AWS Glue:

- Used for ETL processes, including data cleaning, transformation, and schema enforcement.

3.AWS Athena:

- Serverless query engine for analyzing processed data.

4.AWS Lambda:

- Automates the ETL pipeline by triggering Glue jobs upon file uploads to S3.

Programming Choice

Primary Language: Python

Used for:

- Writing Glue ETL jobs for data transformation.
- Automating the pipeline with Lambda.
- Applying PySpark functions for efficient data processing.

Pipeline Execution

1.Data Upload:

- Uploaded raw files (Comments.csv and Courses_info.csv) to the S3 bucket.

2.Data Transformation:

- Processed raw data into normalized fact and dimension tables using Glue ETL jobs.

3.Data Crawling:

- AWS Glue Crawlers populated metadata tables in the udemy_datawarehouse_etl database.

4.Querying:

- Executed SQL queries in Athena to extract insights.

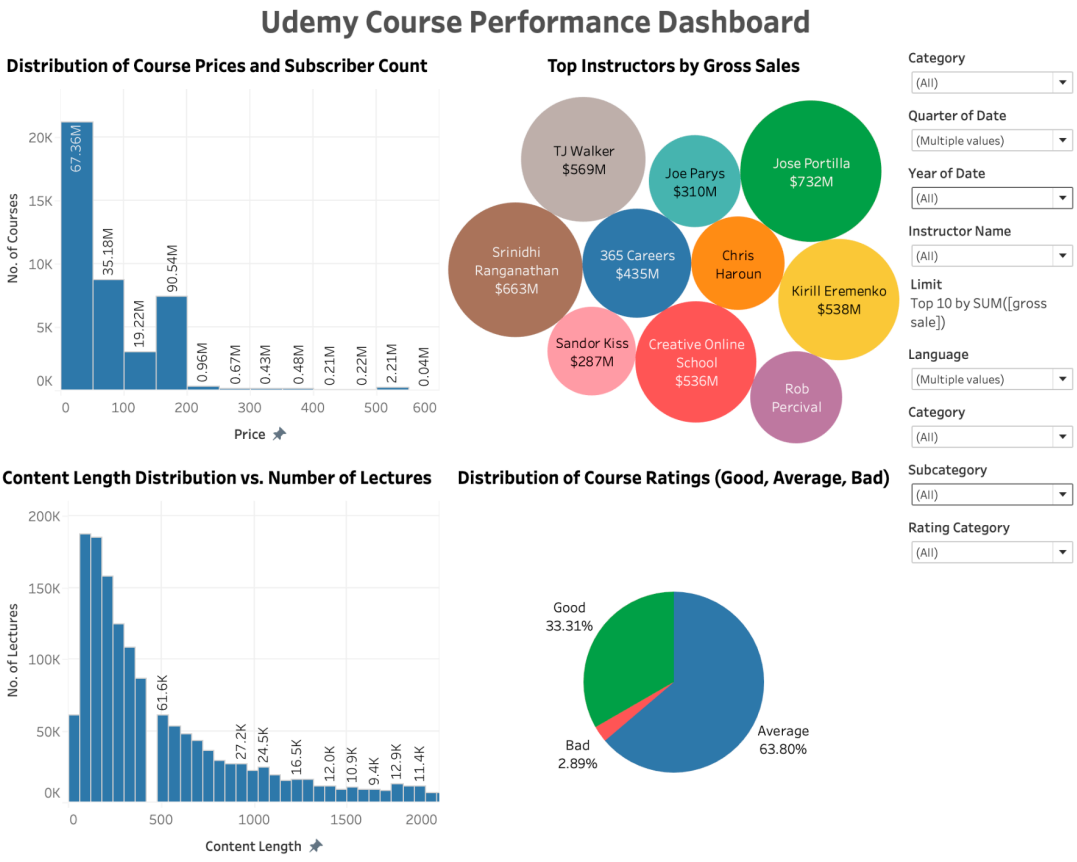
5.Automation:

- Lambda function triggered the Glue ETL job upon new data uploads.

Results and Visualizations

Overview:

The Udemy Course Performance Dashboard provides a holistic view of key metrics related to course pricing, content structure, instructor performance, and user feedback. It is designed to help stakeholders understand trends, identify top-performing courses and instructors, and strategize improvements for better learner engagement and satisfaction.

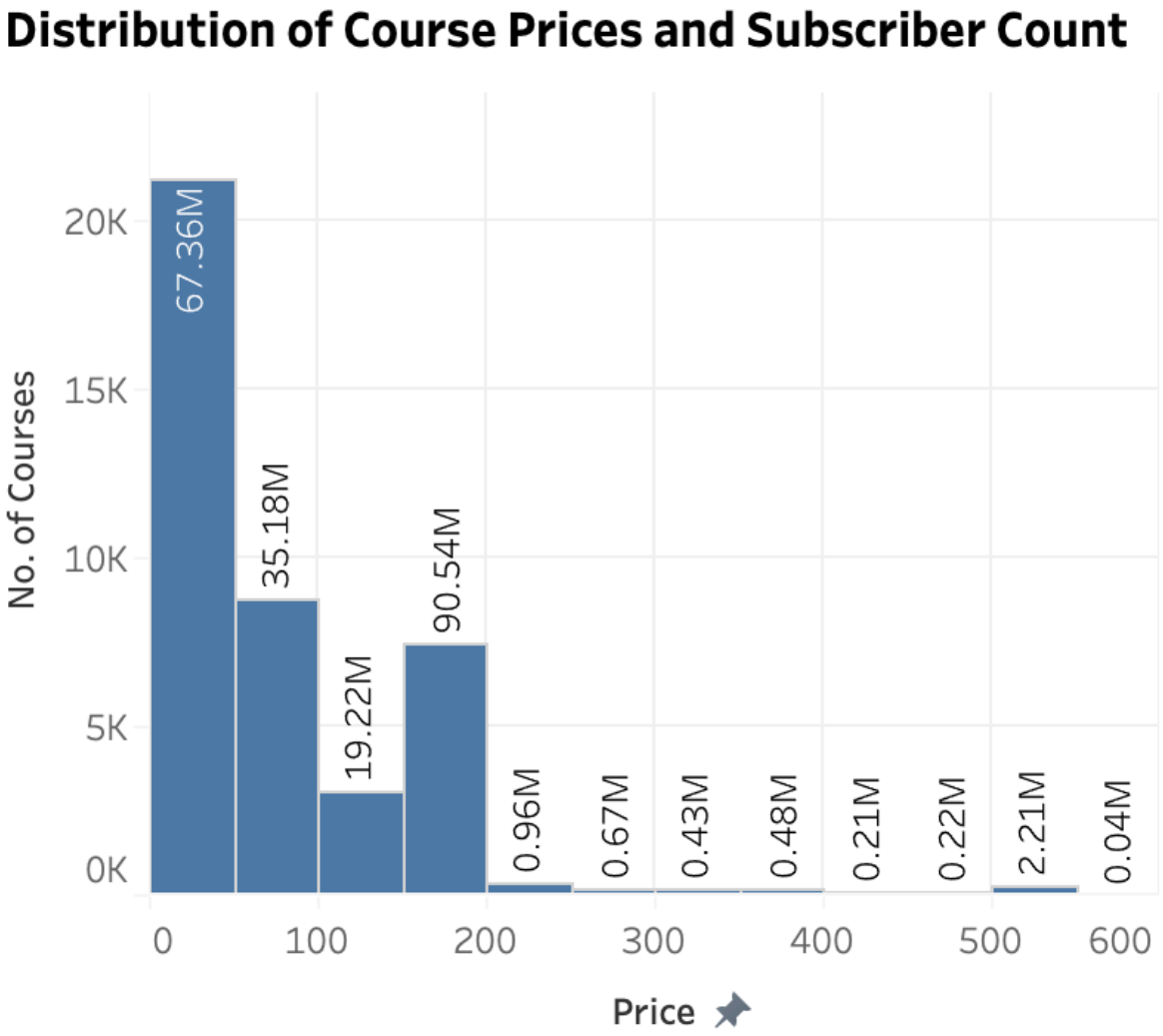


Visualizations, Descriptions, and Insights:

1. Distribution of Course Prices and Subscriber Count (Top Left)

- **Description**: This bar chart displays the distribution of courses across price bins and highlights the total subscriber count in each price range.

- **Purpose**:
  - To analyze how pricing impacts subscriber engagement.
  - To identify the most profitable price ranges.



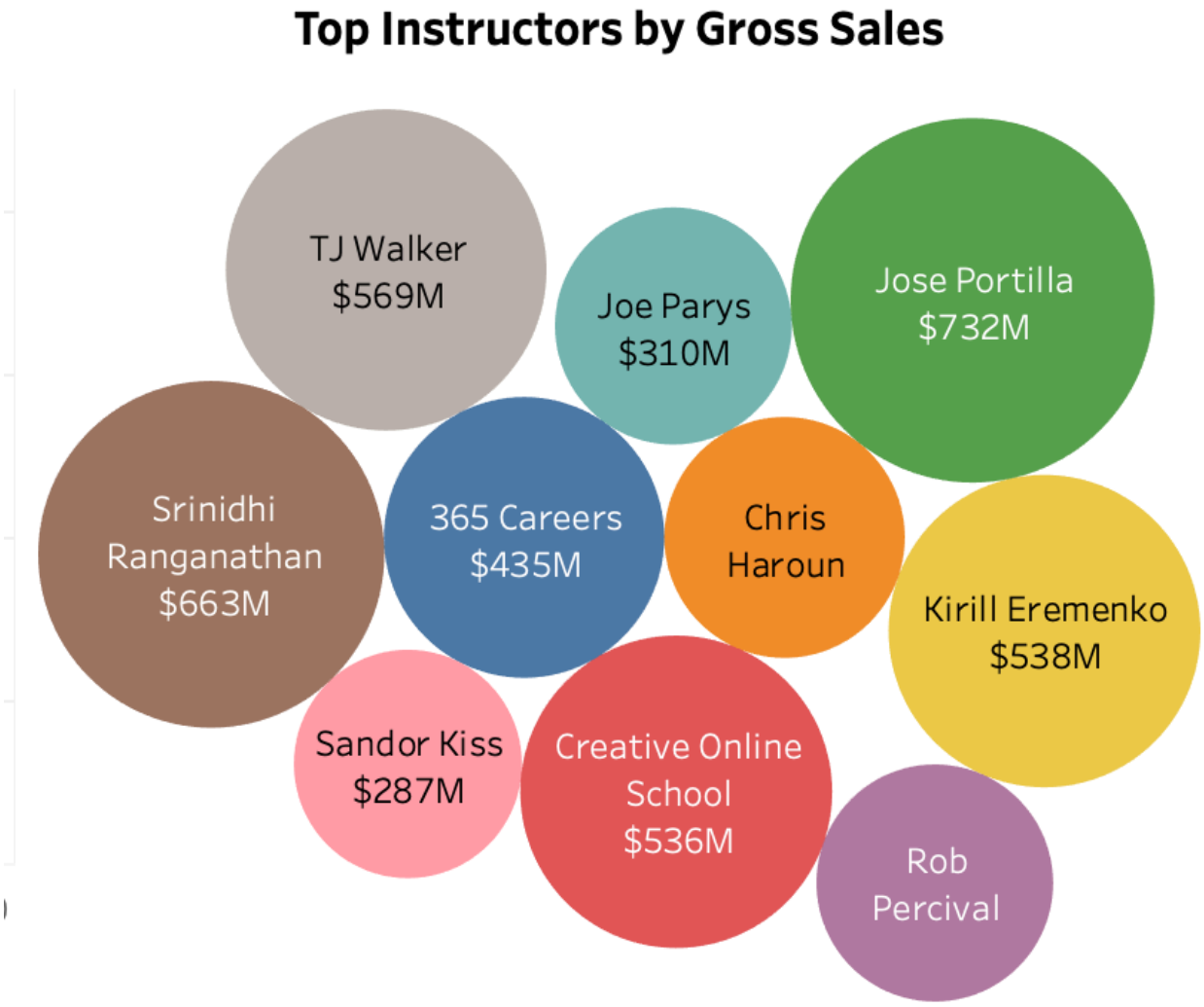
- **KPI**:
  - **Top Price Range for Subscribers**: Price ranges with maximum engagement.
  - **Number of Courses per Price Range**: Pricing patterns across courses.

- **Insights**:
  - A majority of courses are priced under $100, with these courses drawing the highest subscriber numbers.
  - Courses in higher price ranges see significantly fewer subscribers, reflecting price sensitivity among learners.

2. Top Instructors by Gross Sales (Top Right)

- **Description**: A bubble chart showcasing the top instructors based on gross sales. The size of each bubble represents the revenue generated by the instructor.

- **Purpose**:
  - To highlight revenue concentration among top-performing instructors.
  - To identify key contributors to overall revenue.

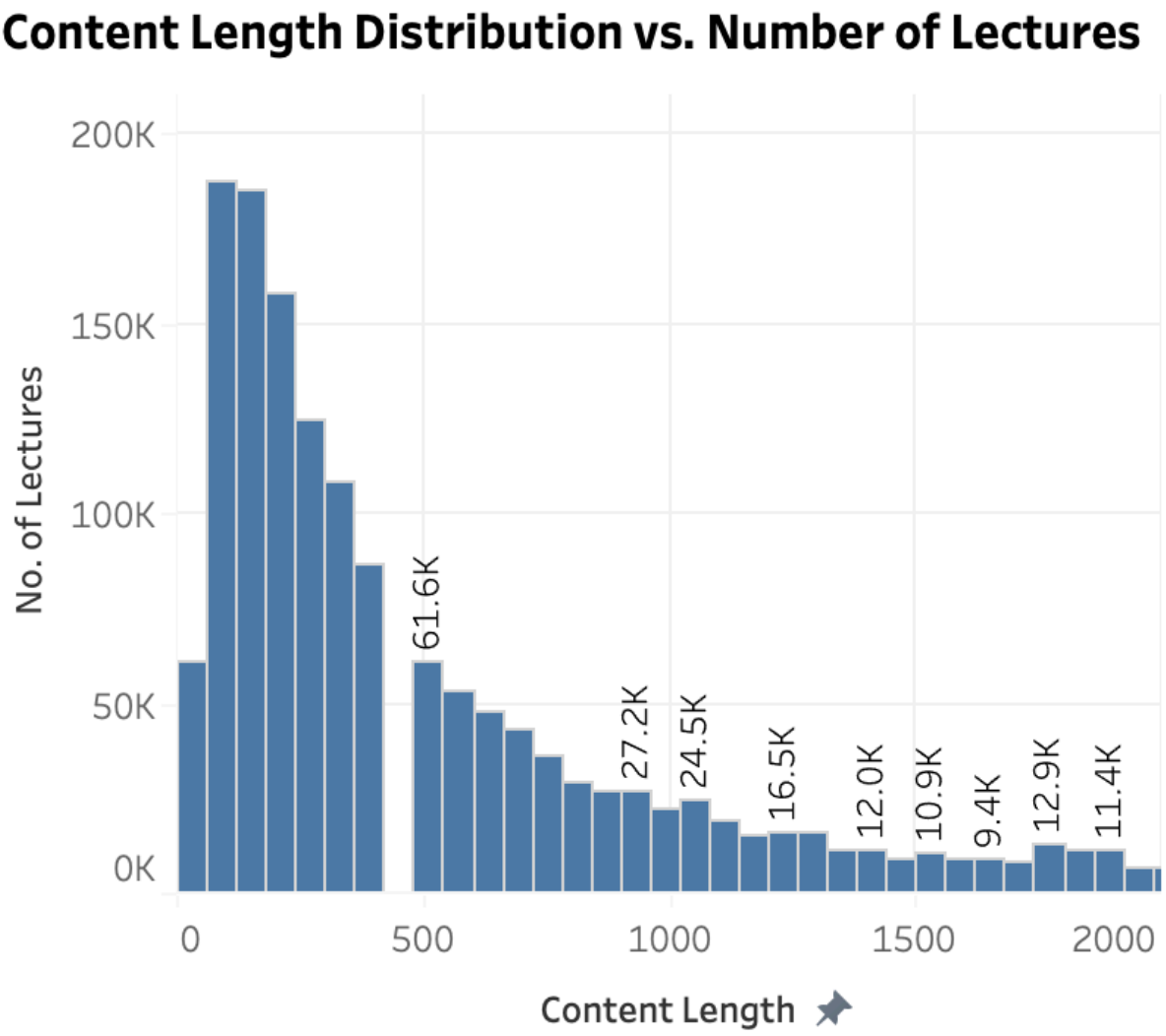


- **KPI**:
  - **Gross Sales by Instructor**: Total revenue generated.
  - **Top Instructors' Market Share**: Contribution of top instructors to total sales.

- **Insights**:
    - "Jose Portilla" leads with $732M in gross sales, followed by "Srinidhi Ranganathan" with $663M, indicating strong individual contributions.
    - Revenue is highly concentrated among the top 10 instructors, suggesting the importance of promoting more diverse contributors.

3. Content Length Distribution vs. Number of Lectures (Bottom Left)

- **Description**: This bar chart shows the relationship between content length (in minutes) and the number of lectures offered in courses.
- **Purpose**:
    - To determine common course durations.
    - To evaluate how content length correlates with course structure.



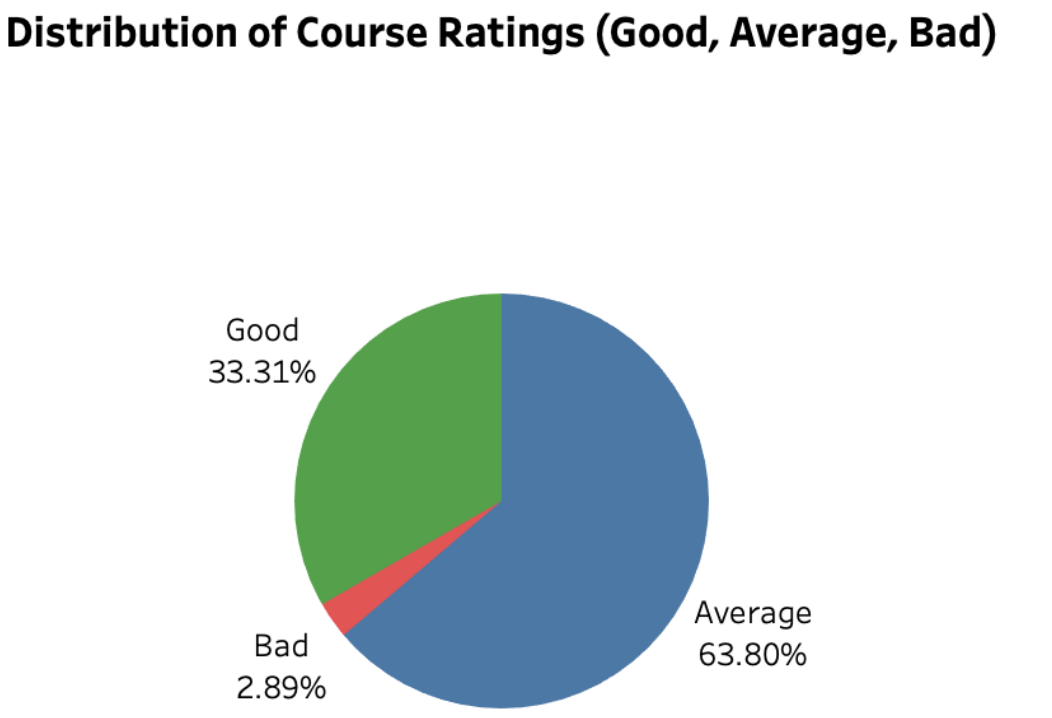
- **KPI**:
    - **Optimal Content Length**: Most frequent duration for courses.
    - **Lecture Count Variance**: Understanding how course length translates into lecture count.

- **Insights**:
    - Courses under 200 minutes are the most common, aligning with learner preference for concise content.
    - Courses with longer durations (>1000 minutes) are significantly fewer, suggesting demand for shorter courses.

4. Distribution of Course Ratings (Good, Average, Bad) (Bottom Right)

- **Description**: A pie chart displaying the proportion of courses categorized as Good, Average, or Bad based on user ratings.

- **Purpose**:
    - To assess the overall quality of courses offered.
    - To identify areas for quality improvement.



- **KPI**:
    - **Percentage of Good Courses**: Proportion of courses rated above 4.5.
    - **Proportion of Average and Bad Courses**: Highlights quality issues.

Scaling:

- **Good (4.5 and above)**: Courses with excellent ratings based on learner satisfaction.
- **Average (3.0 - 4.49)**: Courses with moderate ratings, indicating room for improvement.
- **Bad (Below 3.0)**: Courses with poor ratings requiring immediate attention.

- **Insights**:
    - 63.8% of courses are rated as Average, suggesting significant room for improvement.
    - Only 33.3% of courses are rated as Good, indicating a need to elevate overall course quality.

Filters Included:

1. **Category**: Enables filtering by main categories (e.g., IT, Business).
2. **Quarter and Year of Date**: Allows time-based analysis.
3. **Instructor Name**: Focuses on specific instructor performance.
4. **Subcategory**: Breaks down insights within specific subcategories.
5. **Language**: Compares course performance across different languages.
6. **Rating Category**: Filters courses based on Good, Average, or Bad ratings.

Overall Insights:

1. **Affordable Courses Drive Engagement**: Courses priced under $100 dominate in terms of subscriber count.
2. **Concentration Among Top Instructors**: Revenue is concentrated among the top-performing instructors, highlighting the need for broader instructor empowerment.
3. **Preference for Shorter Content**: Learners favor shorter courses with concise, well-structured material.

4. **Room for Quality Improvement**: A majority of courses fall into the Average rating category, presenting an opportunity to enhance course quality and learner satisfaction.

Future Scoping

This project lays a strong foundation for Udemy's data analytics capabilities, with several avenues for future enhancement:

1. **Advanced Predictive Analytics**:
    - Implement machine learning models to predict course success, optimal pricing, and user satisfaction trends.
2. **Personalized Recommendations**:
    - Integrate user behavior data to recommend courses tailored to individual preferences, increasing learner engagement.
3. **Global Trends Analysis**:
    - Expand data integration to include regional insights, enabling Udemy to target specific markets effectively.
4. **Real-Time Monitoring**:
    - Enhance the pipeline to provide real-time dashboards for tracking course performance and user engagement.
5. **Sentiment Analysis**:
    - Use natural language processing (NLP) techniques to analyze user comments for sentiment insights, helping instructors improve course quality.
6. **Scalability**:
    - Transition to a multi-cloud architecture for enhanced performance and global scalability.
7. **Instructor Insights**:
    - Develop additional KPIs to evaluate instructor improvement over time, based on user feedback and course updates.
8. **Revenue Optimization**:
    - Use advanced analytics to identify pricing elasticity and promotional strategies for maximizing revenue.

Conclusion

The Udemy Course Analysis Project successfully addressed the challenges of unstructured and fragmented data by implementing a comprehensive and automated data pipeline. Using AWS services, the project transformed raw data into structured fact and dimension tables stored in a scalable S3-based data warehouse. Key insights derived from this data enabled Udemy to:

1. Identify optimal pricing strategies and revenue trends.
2. Highlight top-performing instructors and their contributions to the platform.
3. Analyze content length and structure to align with learner preferences.
4. Evaluate course quality through detailed rating and feedback analysis.

The Tableau-based Udemy Course Performance Dashboard serves as a powerful visualization tool for stakeholders, offering interactive filters and dynamic KPIs to support data-driven decision-making. This scalable architecture ensures adaptability for future data growth, providing Udemy with a competitive edge in the dynamic e-learning industry.
