

# **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.

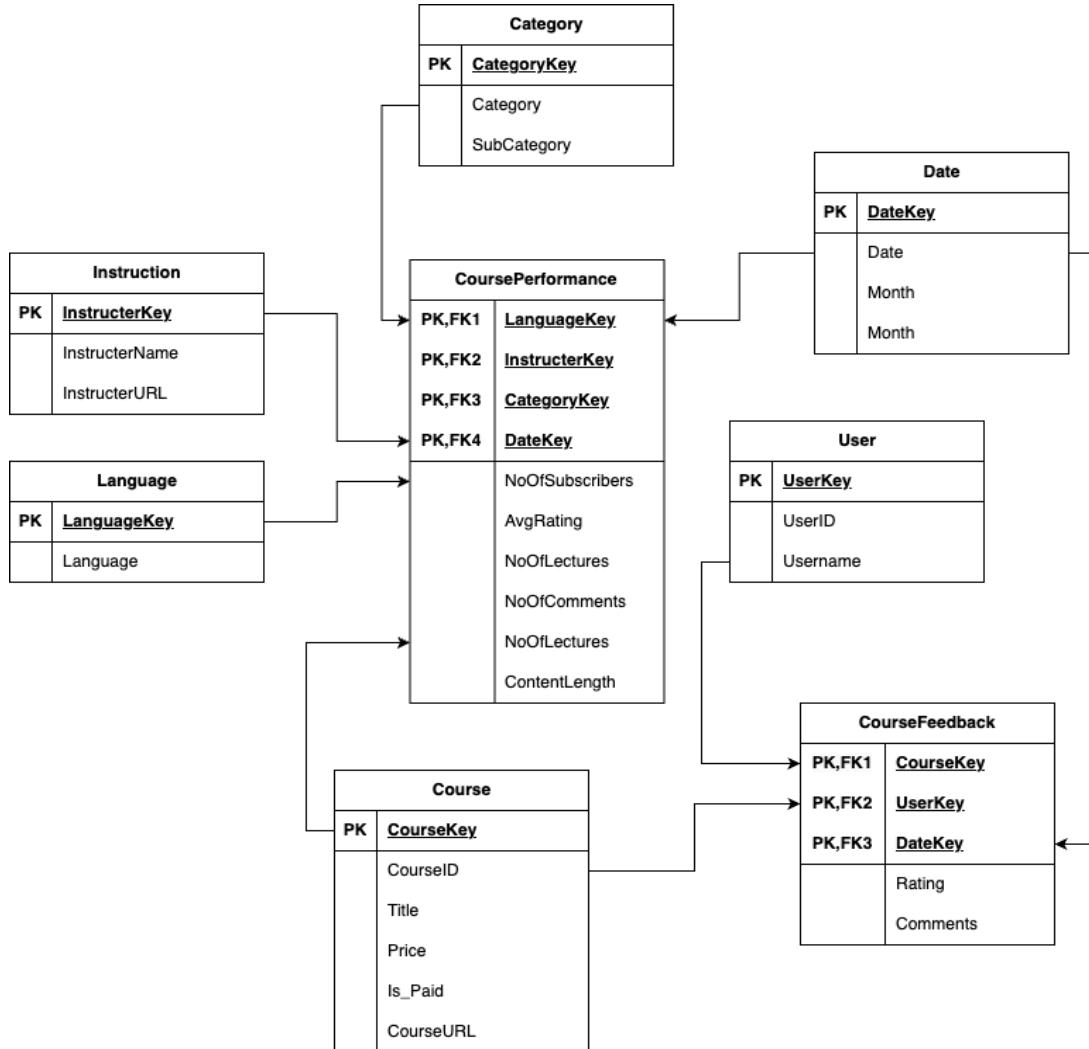
These issues hinder the platform's ability to evaluate course quality, optimize pricing, and support instructors in improving their content. A scalable data warehouse and visualization system are required to address these challenges.

## **Project Objectives**

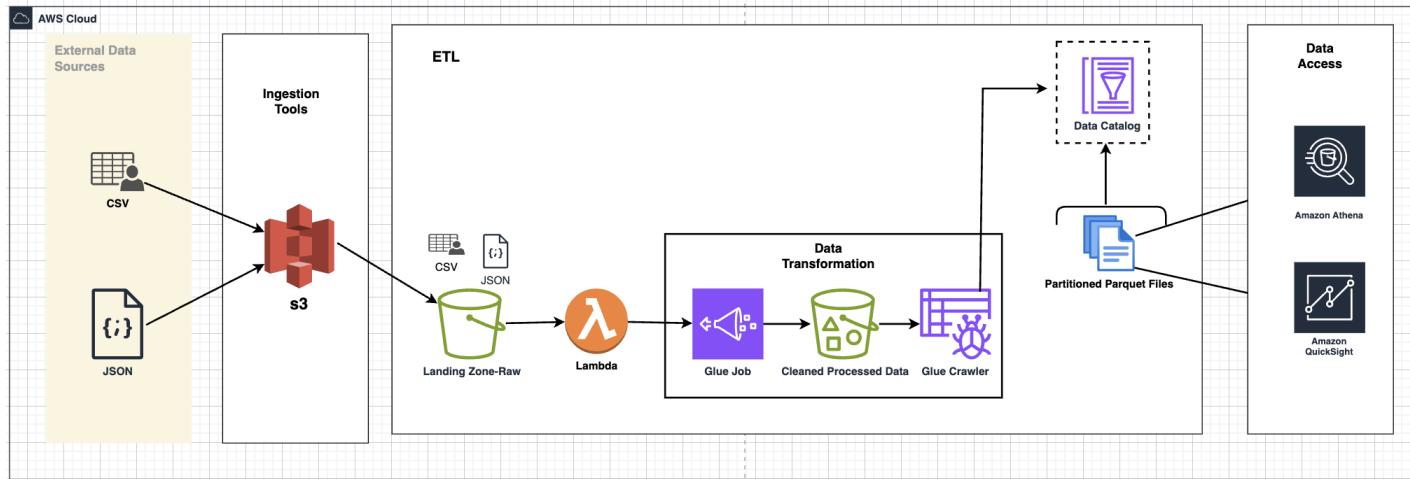
The primary objective of this project is to create an automated data pipeline and a data warehouse for Udemy course analysis. Specific goals include:

- 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

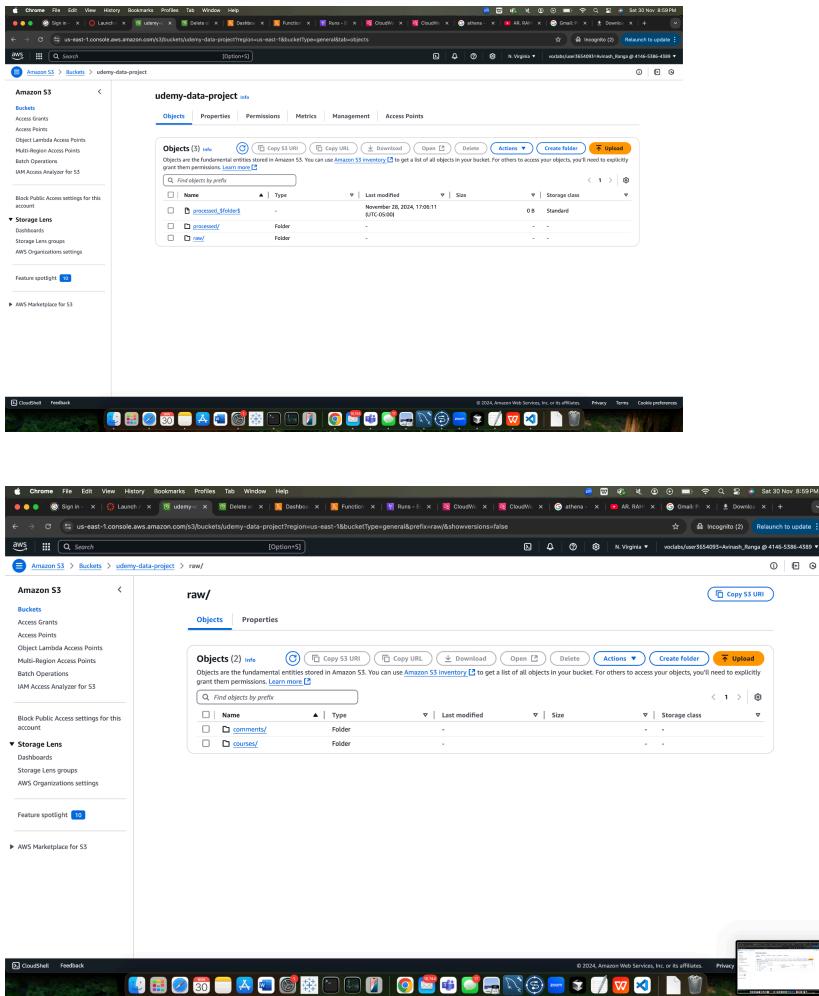
#### Source Files:

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.

```

AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
▼ Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings
Data Integration and ETL
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Interactive Sessions
Data classification tools
Sensitive data detection
Record Matching
Triggers
Workflows (orchestration)
Blueprints
Security configurations
Cost management New
Legacy pages
CloudShell Feedback
aws Search [Option+S] Last modified on 30/11/2024, 16:13:22 Stop notebook Download Notebook Actions Save Run
Notebook Script Job details Runs Data quality Schedules Version Control Upgrade analysis - preview
course_dim = course_df.select(
    col("course_id"),
    col("title"),
    col("price"),
    col("is_paid"),
    col("course_url")
).dropDuplicates()

# Generate sequential IDs
window_spec_course = Window.orderBy("course_id")
course_dim = course_dim.withColumn(
    "course_dim_id", row_number().over(window_spec_course)
)

course_dim.show()

```

course_id	title	price	is_paid	course_url	course_dim_id
2762	Simple Strategy f...	39.99	true	/course/wing-trai...	1
4735	Online Vegan Vegc...	24.99	true	/course/vegan-vegg...	2
5607	How to Train Your ...	19.99	false	/course/training-ho...	3
7723	How to Train a Pupp...	199.99	true	/course/complete-...	4
8867	Curse Me Now - 1349...	1349.99	true	/course/curse-me-n...	5
8879	How to Create an AI...	1349.99	true	/course/ai-creati...	6
8882	Ruby Programming ...	74.99	true	/course/learn-rub...	7
8332	Java Programming ...	19.99	true	/course/learn-java...	8
8157	Web Design From t...	159.99	true	/course/web-design...	9
8316	Java Programming ...	19.99	true	/course/learn-java...	10
8319	Git Basics: In Th...	19.99	true	/course/git-basics...	11
8324	JavaScript for Be...	19.99	true	/course/beginning-j...	12
8404	Python for Data Sc...	19.99	true	/course/python-data...	13
8416	Beginners - How T...	49.99	true	/course/beginners-...	14
8420	Machine Learning F...	49.99	true	/course/machine-lear...	15
8422	Kundalini Yoga fo...	49.99	true	/course/kundalini-y...	16
8423	The Lean Startup   39...	39.99	true	/course/the-lean-sta...	17

## Instructor Dimension: Details about instructors.

```

AWS Glue
Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)
▼ Data Catalog
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings
Data Integration and ETL
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Interactive Sessions
Data classification tools
Sensitive data detection
Record Matching
Triggers
Workflows (orchestration)
Blueprints
Security configurations
Cost management New
Legacy pages
CloudShell Feedback
aws Search [Option+S] Last modified on 30/11/2024, 16:15:52 Stop notebook Download Notebook Actions Save Run
Notebook Script Job details Runs Data quality Schedules Version Control Upgrade analysis - preview
from pyspark.sql.functions import row_number
window_spec_instructor = Window.orderBy("instructor_name", "instructor_url")
instructor_dim = instructor_dim.withColumn(
    "instructor_dim_id", row_number().over(window_spec_instructor)
)
instructor_dim.show()

```

instructor_name	instructor_url	instructor_dim_id
[Redacted]	/user/syntel-A/	1
[Redacted]	/user/nicoleone...	2
[Redacted]	/user/nicktausaint/	3
[Redacted]	/user/murakami-yo.../	4
[Redacted]	/user/teachwithcode/	5
[Redacted]	/user/relixromanh...	6
[Redacted]	/user/schoolsteps...	7
[Redacted]	/user/teachwithcode/	8
[Redacted]	/user/grupoplaneta/	9
[Redacted]	/user/the-arabic-onl...	10
[Redacted]	/user/123yletmeuse...	11
[Redacted]	/user/123yletmeuse...	12
[Redacted]	/user/1itchiscoopr...	13
[Redacted]	/user/22-lucifer/	14
[Redacted]	/user/247-learning/	15
[Redacted]	/user/29-indigo-a...	16
[Redacted]	/user/3d-animation/	17
[Redacted]	/user/36careers/	18
[Redacted]	/user/mesoniquez2/	19
[Redacted]	/user/730-pop-stud...	20

only showing top 20 rows

```

[48]: from pyspark.sql.functions import lit

```

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

```

# AWS Glue Data Catalog Notebook
# Generate sequential IDs for category dimension
category_dim = pd.read_sql_query("SELECT * FROM category", connection)
category_dim = category_dim.withColumn("category_dim_id", F.col("category") + "_" + F.col("subcategory") + "_" + F.col("topic"))
category_dim = category_dim.dropDuplicates()

# Generate sequential IDs for language dimension
window_spec_language = Window.orderBy("language")
language_dim = language_dim.withColumn("language_dim_id", F.col("language") + "_" + F.col("display_name"))
language_dim = language_dim.dropDuplicates()

# Show the generated dimensions
category_dim.show()
language_dim.show()

```

The screenshot shows the AWS Glue Data Catalog interface with a notebook titled "udemy\_etl\_job". The notebook contains Python code using PySpark to generate sequential IDs for categories and languages. The output of the "category\_dim.show()" command is displayed as a table:

category	subcategory	topic	category_dim_id
Business	Business Analytics	AI Testing	1
Business	Business Analytics	Accounting	2
Business	Business Analytics	Algorithms	3
Business	Business Analytics	Artificial	4
Business	Business Analytics	Amazon AWS	5
Business	Business Analytics	Apache	6
Business	Business Analytics	Apache Spark	7
Business	Business Analytics	Apache Kafka	8
Business	Business Analytics	Apache Storm	9
Business	Business Analytics	Analytics	10
Business	Business Analytics	[BPM Business Proc...	11
Business	Business Analytics	[BPMN Business Proc...	12
Business	Business Analytics	Big Data	13
Business	Business Analytics	Blockchain	14
Business	Business Analytics	Business Analysis	15
Business	Business Analytics	Business Intelig...	16
Business	Business Analytics	Business Model	17
Business	Business Analytics	Business Process	18

The output of the "language\_dim.show()" command is displayed as a table:

language	display_name	language_dim_id
Afrikaans	Afrikaans	1
Albanian	Albanian	2
Arabic	Arabic	3
Azerbaijani	Azerbaijani	4
Bengali	Bengali	5
Bulgarian	Bulgarian	6
Burmese	Burmese	7
Catalan	Catalan	8
Croatian	Croatian	9
Czech	Czech	10
Danish	Danish	11
Dutch	Dutch	12
English	English	13
Esperanto	Esperanto	14
Filipino	Filipino	15
Finnish	Finnish	16
French	French	17
German	German	18
Greek	Greek	19
Hebrew	Hebrew	20

**Language Dimension:** Tracks the course's language.

```

# AWS Glue Data Catalog Notebook
# Generate sequential IDs for language dimension
language_dim = pd.read_sql_query("SELECT * FROM language", connection)
language_dim = language_dim.withColumn("language_dim_id", F.col("language") + "_" + F.col("display_name"))
language_dim = language_dim.dropDuplicates()

# Show the generated dimension
language_dim.show()

```

The screenshot shows the AWS Glue Data Catalog interface with a notebook titled "udemy\_etl\_job". The notebook contains Python code using PySpark to generate sequential IDs for languages. The output of the "language\_dim.show()" command is displayed as a table:

language	display_name	language_dim_id
Afrikaans	Afrikaans	1
Albanian	Albanian	2
Arabic	Arabic	3
Azerbaijani	Azerbaijani	4
Bengali	Bengali	5
Bulgarian	Bulgarian	6
Burmese	Burmese	7
Catalan	Catalan	8
Croatian	Croatian	9
Czech	Czech	10
Danish	Danish	11
Dutch	Dutch	12
English	English	13
Esperanto	Esperanto	14
Filipino	Filipino	15
Finnish	Finnish	16
French	French	17
German	German	18
Greek	Greek	19
Hebrew	Hebrew	20

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

The screenshot shows a Chrome browser window with the AWS Glue Data Catalog notebook titled "udemy\_etl\_job". The notebook contains a PySpark script for generating a surrogate key for users and displaying the User Dimension table.

```

# Generate surrogate key for users
window_spec_user = Window.orderBy("user_id")
user_dim = user_dim.withColumn(
    "user_did_ud", row_number().over(window_spec_user)
)

# Display the User Dimension table
user_dim.show()

```

The output shows the User Dimension table with columns [user\_id] and [display\_name|user\_did]. The data includes:

[user_id]	[display_name user_did]
25514	am
41986	"mel darling"
71658	fernando
76400	avast!
93122	christy
108952	denirini.angelabagh...
118100	rui
133166	sergio
147182	tony
150000	wilma
182512	alicia
187928	karen.pudephatt@...
199158	d
224000	isabel
226622	david
243164	subhasis
247038	maria
247038	theresa
248058	dave
248398	laura

only showing top 28 rows

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

The screenshot shows a Chrome browser window with the AWS Glue Data Catalog notebook titled "udemy\_etl\_job". The notebook contains a PySpark script for generating a date dimension table.

```

date_dim = unique_dates.withColumn("year", year(col("date")))
date_dim = date_dim.withColumn("month", month(col("date")))
date_dim = date_dim.withColumn("day", dayOfMonth(col("date")))
date_dim = date_dim.withColumn("date_did", row_number().over(Window.orderBy("date")))

# Generate sequential IDs
window_spec_date = Window.orderBy("date")
date_dim = date_dim.withColumn(
    "date_did_ud", row_number().over(window_spec_date)
)

date_dim.show()

```

The output shows the Date Dimension table with columns [date|year|month|day|date\_din\_ud]. The data includes:

[date year month day date_din_ud]
2018-04-14 2018 4 14 1
2018-05-05 2018 5 5 2
2018-05-05 2018 5 5 3
2018-10-13 2018 10 13 4
2019-01-01 2019 1 1 5
2011-06-23 2011 6 23 6
2011-07-01 2011 7 1 7
2011-07-01 2011 7 1 8
2011-07-09 2011 7 9 9
2011-07-16 2011 7 16 10
2011-07-17 2011 7 17 11
2011-07-15 2011 7 15 12
2011-07-18 2011 7 18 13
2011-07-21 2011 7 21 14
2011-07-28 2011 7 28 15
2011-07-29 2011 7 29 16
2011-08-01 2011 8 1 17
2011-08-08 2011 8 8 18
2011-08-13 2011 8 13 19
2011-08-22 2011 8 22 20

only showing top 20 rows

## Fact Table Creation:

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

```

# Display the CoursePerformance Fact Table
course_performance_fact.show()

```

course_dim_id	language_dim_id	category_dim_id	instructor_dim_id	published_date_dim_id	num_subscribers	avg_rating	num_reviews	num_comments	num_lectures	content_length
26.0	6	13	1231	18654	7	18761.0	3.9	349.0	181.0	87.0
13.0	15	13	5958	2885	14	4454.0	4.35	829.0	147.0	238.0
48.0	53	13	1670	2136	48	743.0	4.3	87.0	22.0	33.0
70.0	68	13	6798	8988	45	32813.0	4.18795	2824.0	987.0	44.0
78.0	82	13	6796	18139	55	582.0	4.3	42.0	19.0	19.0
37.0	93	13	1869	2855	62	5825.0	4.196429	604.0	126.0	98.0
34.0	96	13	9259	1962	125	15652.0	4.681818	1913.0	648.0	54.0
68.0	110	13	3920	13822	70	78.0	5.0	6.0	2.0	18.0
58.0	138	13	7743	6569	84	1828.0	4.55	122.0	48.0	75.0
38.0	179	13	2232	14737	166	2911.0	3.9	92.0	24.0	55.0
68.0	205	13	6885	13822	117	131.0	4.7	24.0	19.0	65.0
55.0	206	13	2381	662	113	2348.0	3.55	19.0	5.0	55.0

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

```

.select(
    col("course_dim_id"),
    col("user_dim_id"),
    col("feedback_date_dim_id").alias("rate"),
    col("rate"),
    col("comment")
)

```

```

# Step 4: Show result
course_feedback_fact.show()

```

course_dim_id	user_dim_id	feedback_date_dim_id	rate	comment
443	33	1296	5.0	[...]When I signed up [...]
80	49	1517	5.0	[...]I enjoyed the last video [...]
586	10	739	5.0	[...]Thank you for this course!
1051	1	483	5.0	[...]I have been involved in [...]
1852	39	1358	5.0	[...]Meta is a wonderful [...]
1851	1880	3075	5.0	[...]I am very happy with this course!
1382	22	1113	4.0	[...]very good course! I liked it!
1119	6	585	5.0	[...]This is a solid [...]
1251	123	2092	5.0	[...]I really enjoyed this course! [...]
1283	91	671	5.0	[...]I found this course [...]
1353	191	2387	2.0	[...]corso è molto [...]
1353	2	467	5.0	[...]great with tons of [...]
1321	288	2377	5.0	[...]I really like this course!
1563	6807	3732	3.0	[...]Average, I think [...]
1651	31	524	5.0	[...]Very well explained![...]
1714	9815	8097	5.0	[...]I liked it!
1714	5	538	5.0	[...]Awesome course in [...]
1733	41	530	4.0	[...]Me ulevade, miss [...]
1839	5227	3645	3.0	[...]Content-wise a go [...]
1844	32	1291	5.0	[...]Great course! Atta [...]

## 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.

Name	State	Last run	Last run timestamp	Log	Table changes from last run
category_dim	Ready	Succeeded	November 30, 2024 at ...	<a href="#">View log</a>	1 created
comments_crawler	Ready	Succeeded	November 30, 2024 at ...	<a href="#">View log</a>	1 created
course_dim	Ready	Succeeded	November 30, 2024 at ...	<a href="#">View log</a>	1 created
course_feedback_fact	Ready	Succeeded	November 30, 2024 at ...	<a href="#">View log</a>	1 created
course_performance_fact	Ready	Succeeded	November 30, 2024 at ...	<a href="#">View log</a>	1 created
courses_crawler	Ready	Succeeded	November 30, 2024 at ...	<a href="#">View log</a>	1 created
date_dim	Ready	Succeeded	November 30, 2024 at ...	<a href="#">View log</a>	-
instructor_dim	Ready	Succeeded	November 30, 2024 at ...	<a href="#">View log</a>	1 created
user_dim	Ready	Succeeded	November 30, 2024 at ...	<a href="#">View log</a>	1 created

## 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.

Sat 30 Nov 9:16 PM

us-east-1.console.aws.amazon.com/athena/home?region=us-east-1#query-editor/history/03614ec1-f8a9-4a86-8dd-60d3b036826f

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.

**Data**

Data source: AwsDataCatalog

Database: udemy\_datadwhouse\_done

Tables and views: category\_dim, course\_dim, course\_feedback\_fact, course\_performance\_fact, date\_dim, instructor\_dim, user\_dim

Tables (7): category\_dim, course\_dim, course\_feedback\_fact, course\_performance\_fact, date\_dim, instructor\_dim, user\_dim

Views (0)

**Query 9**: SELECT course\_dim\_id, title, price, is\_paid, course\_url FROM course\_dim LIMIT 10;

SQL Ln 1, Col 59

**Run Explain Cancel Clear Create**

**Query results** | **Query stats**

Completed Time in queue: 106 ms Run time: 533 ms Data scanned: 2.79 MB

**Results (10)**

#	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

Sat 30 Nov 9:16 PM

us-east-1.console.aws.amazon.com/athena/home?region=us-east-1#query-editor/history/0954a3cfb-ba87-4e59-a29e-67198a5fb3f

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.

**Data**

Data source: AwsDataCatalog

Database: udemy\_datadwhouse\_done

Tables and views: category\_dim, course\_dim, course\_feedback\_fact, course\_performance\_fact, date\_dim, instructor\_dim, user\_dim

Tables (7): category\_dim, course\_dim, course\_feedback\_fact, course\_performance\_fact, date\_dim, instructor\_dim, user\_dim

Views (0)

**Query 10**: -- the most expensive courses

```
1 -- the most expensive courses
2 SELECT title, price
3   FROM course_dim
4 ORDER BY price DESC
5 LIMIT 10;
```

SQL Ln 5, Col 25

**Run again Explain Cancel Clear Create**

**Query results** | **Query stats**

Completed Time in queue: 112 ms Run time: 809 ms Data scanned: 1.52 MB

**Results (10)**

#	title	price
1	Entity Framework Eğitim Videosu Serisi	999.99
2	C# Programlama Dili : Temel, Orta, İleri Seviye	999.99

CloudShell Feedback

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.

**Data source**: AwsDataCatalog  
**Database**: udemy\_datawarehouse\_done

**Tables and views**: [Create](#)

**Tables** (7): category\_dim, course\_dim, course\_feedback\_fact, course\_performance\_fact, date\_dim, instructor\_dim, user\_dim

**Views** (0)

```

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](#)

**Query results** | **Query stats**

**Completed** Time in queue: 107 ms Run time: 460 ms Data scanned: 0.37 KB

**Results (1)**

#	is_paid	course_count
1	true	44673

[Copy](#) [Download results](#)

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.

**Data source**: AwsDataCatalog  
**Database**: udemy\_datawarehouse\_done

**Tables and views**: [Create](#)

**Tables** (7): category\_dim, course\_dim, course\_feedback\_fact, course\_performance\_fact, date\_dim, instructor\_dim, user\_dim

**Views** (0)

```

1 -- top 10 instructors with the most subscribers.
2 SELECT
3     i.instructor_name,
4     SUM(cp.num_subscribers) AS total_subscribers
5     FROM
6     course_performance_fact cp
7     JOIN
8     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](#)

**Query results** | **Query stats**

**Completed** Time in queue: 104 ms Run time: 1.172 sec Data scanned: 546.86 KB

**Results (10)**

#	instructor_name	total_subscribers
1	Phil Ebner	4982511.0

[Copy](#) [Download results](#)

The screenshot shows the Amazon Athena Query Editor interface. On the left, there's a sidebar for 'Data' with 'Data source' set to 'AwsDataCatalog' and 'Database' set to 'udemy\_datawarehouse\_staging'. Below that is a list of 'Tables and views' under 'Tables (7)'. The main area shows a query editor with multiple tabs (Query 8, 9, 11, 13, 14) and a new tab 'Query 14' which contains the following SQL code:

```

1 --top 10 courses by the number of subscribers, along with their reviews and average rating.
2
3 SELECT
4     c.title AS course_title,
5     cp.num_subscribers,
6     cp.num_reviews,
7     cp.avg_rating
8 FROM
9     course_performance_fact cp
10 JOIN
11     course_dim c
12 ON
13     cp.course_dim_id = c.course_dim_id
14 ORDER BY
15     cp.num_subscribers DESC

```

Below the code, there are buttons for 'Run', 'Explain', 'Cancel', 'Clear', and 'Create'. To the right, there's a 'Reuse query results' link. At the bottom, the 'Query results' tab is selected, showing a table with one row of data:

course_title	num_subscribers	num_reviews	avg_rating
2022 Complete Python Bootcamp From Zero to Hero in Python	1612862.0	456457.0	4.5

Other tabs include 'Query stats' and 'Completed'.

## Key Benefits:

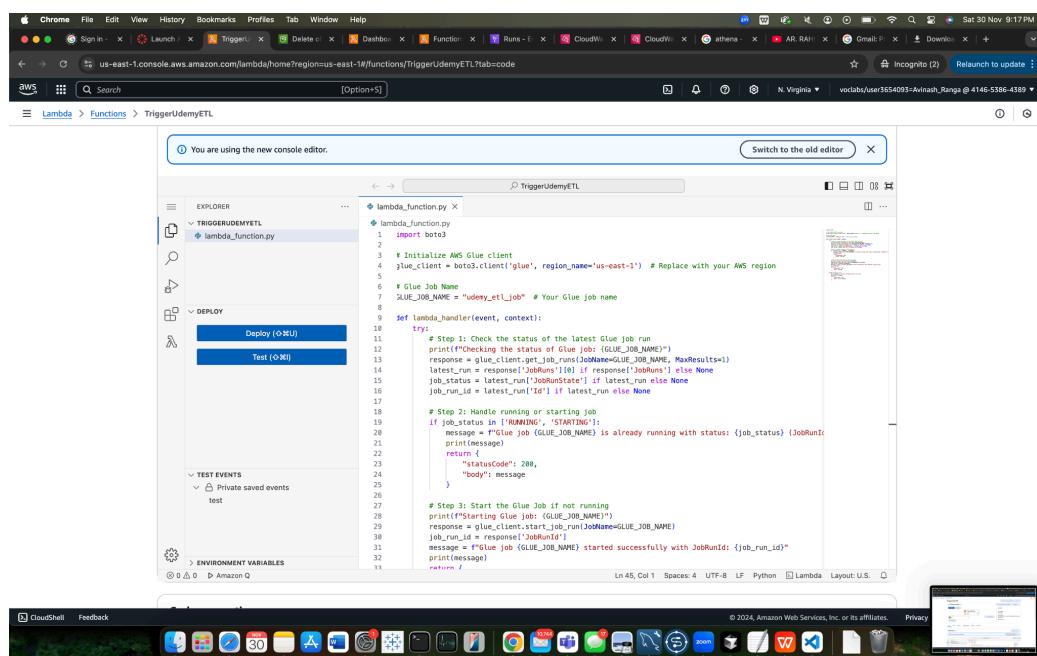
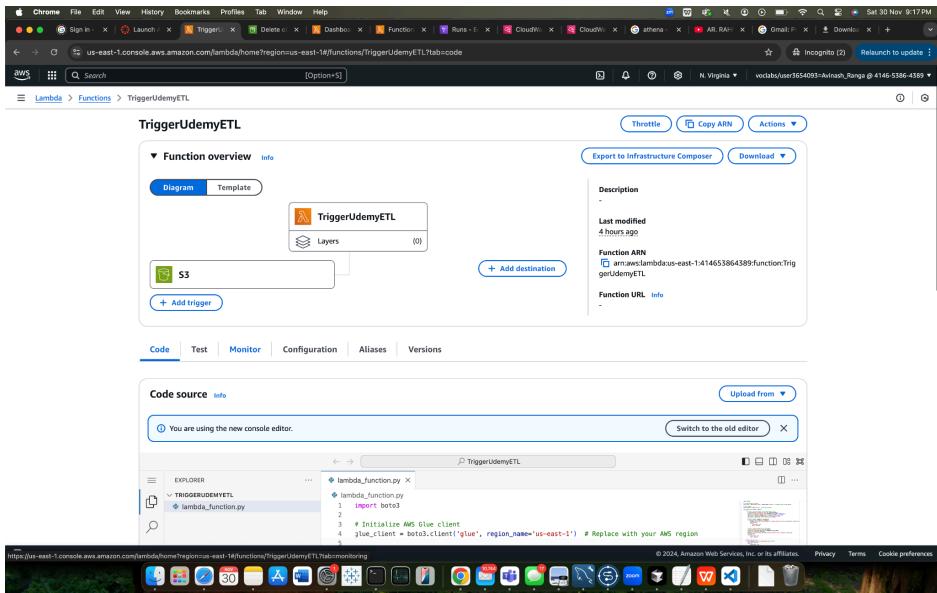
- Serverless query execution with a pay-per-query model.
- Efficient OLAP-style querying over large datasets stored in S3.

## 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.



## 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.

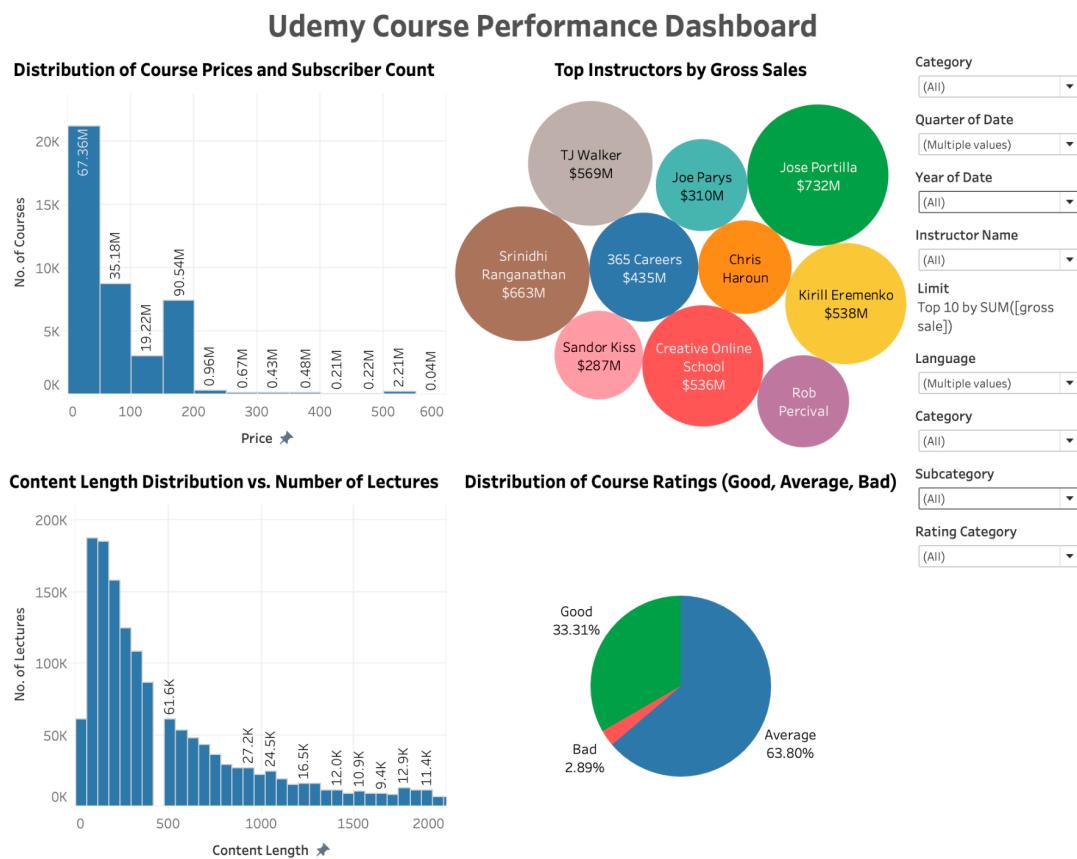
### **5. Automation:**

- 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.



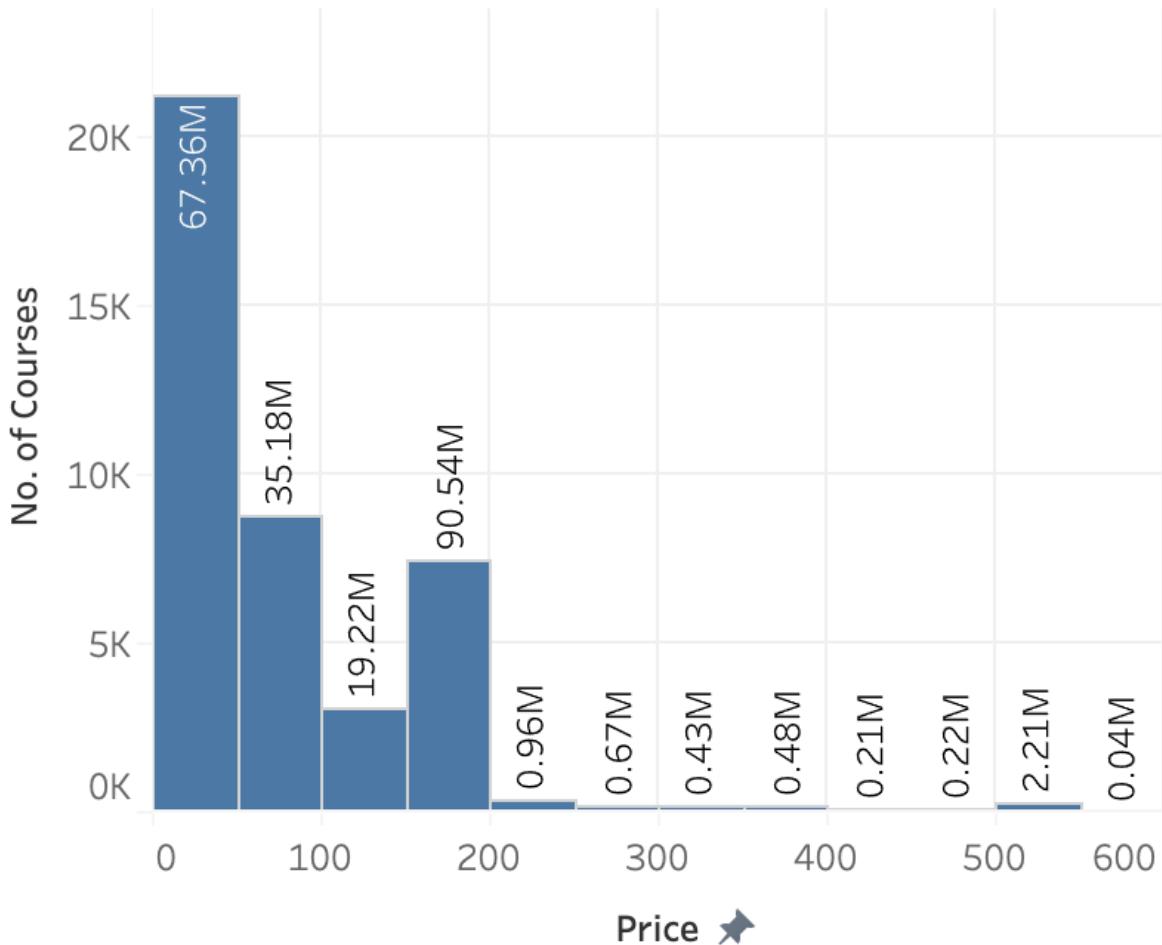
## Visualizations, Descriptions, and Insights:

### 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.

## Distribution of Course Prices and Subscriber Count

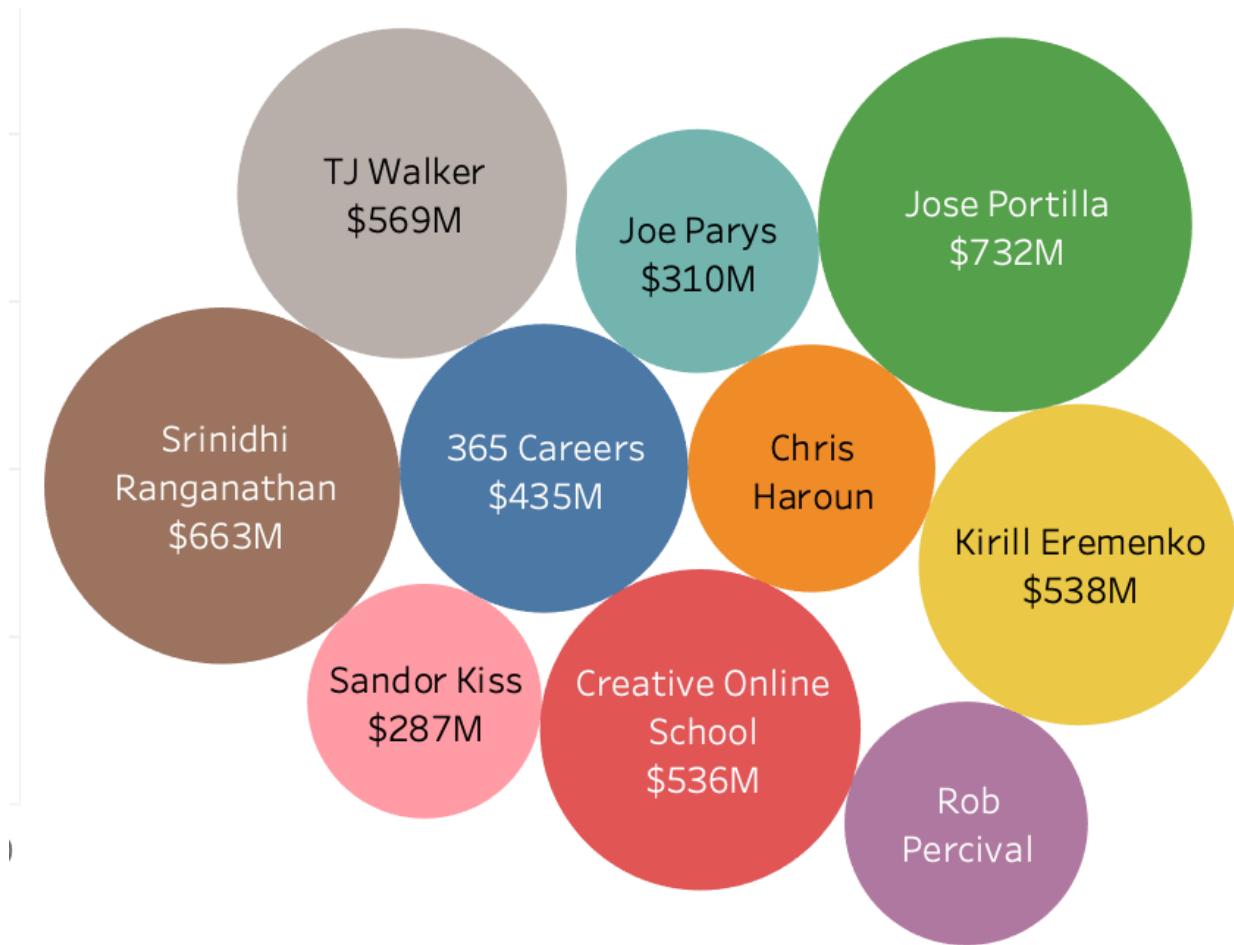


- **KPI:**
  - **Top Price Range for Subscribers:** Price ranges with maximum engagement.
  - **Number of Courses per Price Range:** Pricing patterns across courses.
- **Insights:**
  - A majority of courses are priced under \$100, with these courses drawing the highest subscriber numbers.
  - Courses 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.

### **Top Instructors by Gross Sales**



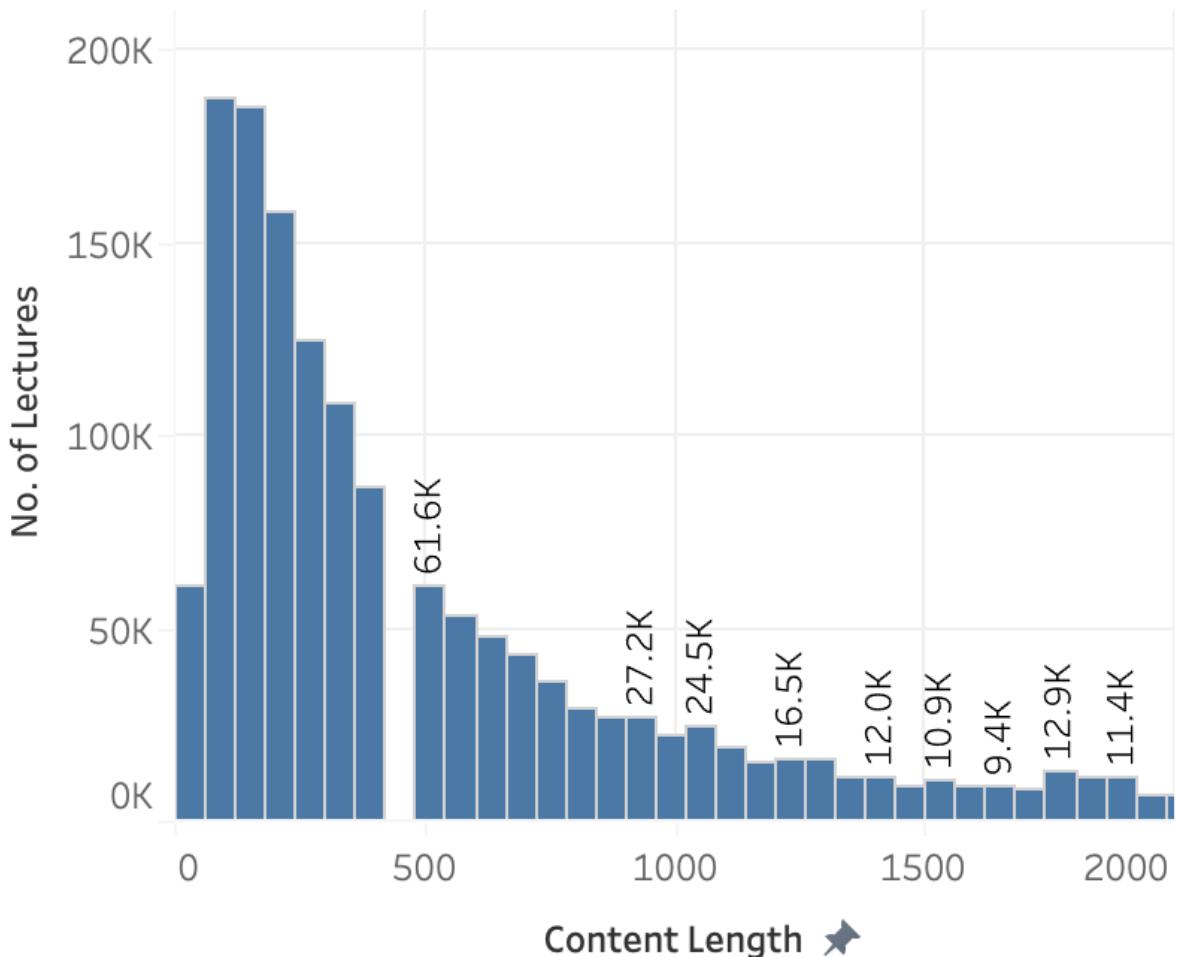
- **KPI:**
  - **Gross Sales by Instructor:** Total revenue generated.
  - **Top Instructors' Market Share:** Contribution of top instructors to total sales.

- **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.

## **Content Length Distribution vs. Number of Lectures**

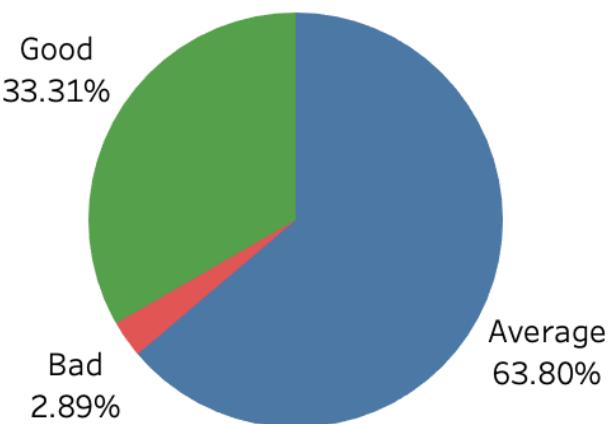


- **KPI:**
  - **Optimal Content Length:** Most frequent duration for courses.
  - **Lecture Count Variance:** Understanding how course length translates into lecture count.
- **Insights:**
  - Courses under 200 minutes are the most common, aligning with learner preference for concise content.
  - Courses 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.

#### **Distribution of Course Ratings (Good, Average, Bad)**



- **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.

---