🎯 Resume Optimization: Transitioning to a PySpark-Focused Data Engineering Role

Your current resume highlights strong cloud migration, BigQuery, and orchestration (Airflow) skills. The objective now is to pivot the narrative to focus on **scalable data processing** and **distributed computing principles**, directly addressing the PySpark requirement.

1. Top Section & Title

Suggested Change

- **Current Title:** Data Engineer
- **Suggested Title:** Data Engineer | Cloud ETL & Big Data Specialist (PySpark Focus)

Rationale

This instantly signals to the recruiter that you have the core skills *plus* the emerging focus area they are searching for, making your resume pass initial screening filters.

2. Skills Section Enhancement (Crucial for PySpark)

The skills section is the first place a recruiter looks for keywords. You need to elaborate on your PySpark learning, even if it's from personal projects.

Suggested Changes

Current Skills Category	Suggested Skill Updates	Rationale
GCP Services	**Dataproc (Intermediate/Applied)**, **Dataflow (Basic)**. *(Add Data Lake / Delta Lake if you have project experience.)*	**Increase the perceived proficiency.** "Dataproc" is where PySpark runs. Change "Basic" to "Intermediate" if you have completed robust projects.
Languages	Python (Advanced), SQL (Advanced), Shell Script, Unix.	Emphasize **Python** as your primary data language.
New Category	**Big Data Frameworks:** Apache Spark (PySpark), Spark SQL.	**Directly address the gap.** This shows you are not just learning, but you are fluent in the framework's terminology.
New Category	**Data Lake Concepts:** Parquet, **Delta Lake** (Crucial if you used it in projects).	Modern PySpark roles require knowledge of columnar file formats and data lake optimization.

ETL Tool	IBM InfoSphere Datastage (Historical/Legacy ETL).	Keep it, but position it as your legacy experience, while emphasizing modern/cloud tools.

3. Experience Section Re-Wording

The goal here is to replace vague statements with **action-oriented results** that emphasize **performance** and **scalability**, which are the core themes of PySpark.

A. Reframing the BigQuery Work (Simulating Distributed Logic)

You have great points on optimization and SCD logic. Rephrase these to use distributed computing language.

Original Bullet Point (BigQuery)	Suggested PySpark-Oriented Re-write	Focus
"I carried out performance tuning and optimization on the migrated data systems to improve query performance and efficiency, resulting in a 30% reduction in system runtime."	"Optimized BigQuery ETL processes using advanced SQL features (**Window Functions, Clustering, and Partitioning**) to achieve a 30% reduction in query runtime and cost efficiency."	**Optimization & Advanced SQL** (skills transferable to Spark SQL).
"Implemented SCD Type 1 and Type 2 logic in BigQuery, ensuring historical data accuracy and enabling incremental load capabilities..."	"Designed and implemented robust **Slowly Changing Dimension (SCD Type 1 & 2)** logic for critical financial datasets, utilizing **MERGE** statements for highly efficient, incremental data updates."	**Data Modeling & MERGE** (Directly transferable to Delta Lake MERGE in Spark).
"Contributed to the migration from DataStage to BigQuery using GCP services, enabling a scalable and efficient cloud-based ETL framework..."	"Led the transition of legacy ETL processes to a scalable **Cloud-Native Data Pipeline**, designing solutions to handle high-volume data streams leveraging Airflow and BigQuery."	**Scalability & Cloud Native** (Emphasizing the architectural shift).

B. Incorporating PySpark Project Experience (The Key Change)

Since you don't have professional PySpark experience, you must add a dedicated section to showcase your portfolio projects. **This replaces the professional gap.**

Add a section directly below your "Datametica Birds" experience, or integrate it as a **"Project Spotlight"** within your professional summary.

Suggested New Section Template:

Applied Data Engineering Projects (PySpark & Cloud)

- **Project 1: Real-Time Fraud Data Pipeline (PySpark/Dataproc/Delta Lake)**
  - **Goal:** Engineered an end-to-end data pipeline to ingest 10GB of simulated financial transaction data, performing feature engineering and writing the results to a structured data

lake.

- **Key Achievement:** Developed PySpark transformations, utilizing **Broadcast Joins** to efficiently link customer reference data to transaction streams, resulting in a 5x speedup compared to standard joins. * **Tools:** PySpark, Spark SQL, Dataproc, GCS, Delta Lake.

- **Project 2: Legacy Data Ingestion Engine (Airflow & PySpark)**
  - **Goal:** Built an Airflow DAG to orchestrate a PySpark job that cleans and standardizes messy CSV files into a quality-controlled Parquet format.
  - **Key Achievement:** Implemented data quality checks and used **Pandas UDFs** for vectorized string operations (like phone number standardization), showcasing best practices for custom logic on Spark.

4. Summary of Strategy

1. **Rephrase, Don't Fabricate:** Use PySpark terminology (**Broadcast Join, Window Functions, MERGE**) to describe your BigQuery/SQL experience, highlighting the transferable skills.
2. **Dedicate Space:** Make your self-learned PySpark skills prominent in a dedicated "Applied Projects" section. This shows initiative and practical application.
3. **Beef up the Skills Section:** Ensure the top section has all the PySpark/Big Data keywords needed to pass the initial automated checks.

By making these changes, your resume will strongly position you as a candidate who possesses core