Firebase-Based Recipe Analytics Pipeline

Technical Report

Executive Summary

This report documents a complete data engineering pipeline built on Firebase Firestore for recipe data extraction, transformation, validation, and analytics. The system processes an authentic Maharashtrian cuisine dataset containing 20+ recipes, demonstrating modern ETL practices with NoSQL databases.

---

1. Introduction

1.1 Project Overview

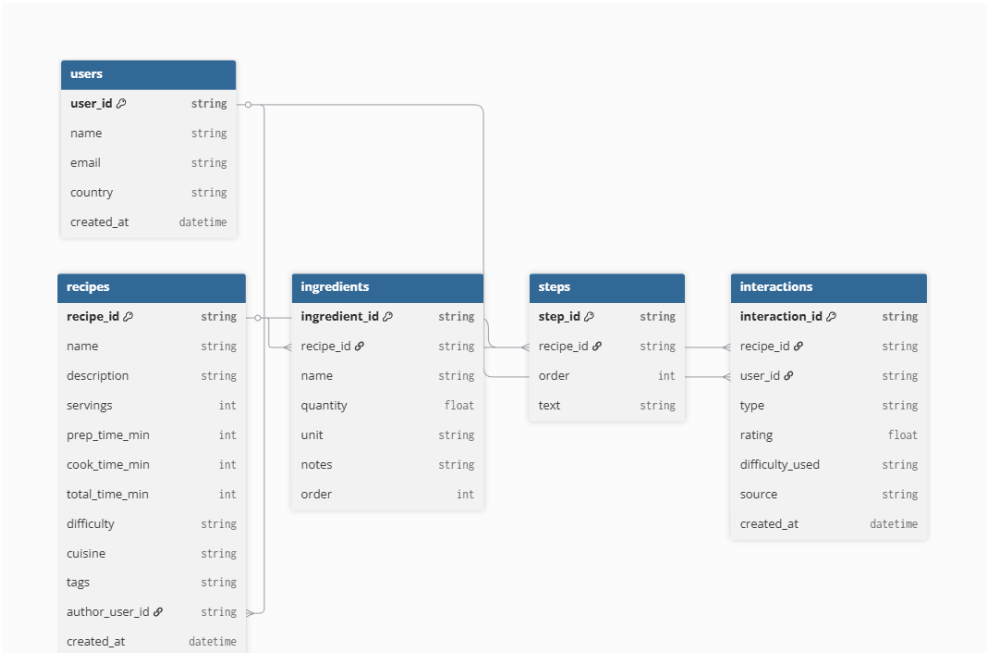
The Recipe Analytics ETL Pipeline is designed to extract recipe data from Firebase Firestore, transform it into normalized relational formats, validate data quality, and generate actionable insights through analytics and visualizations.

1.2 Technology Stack

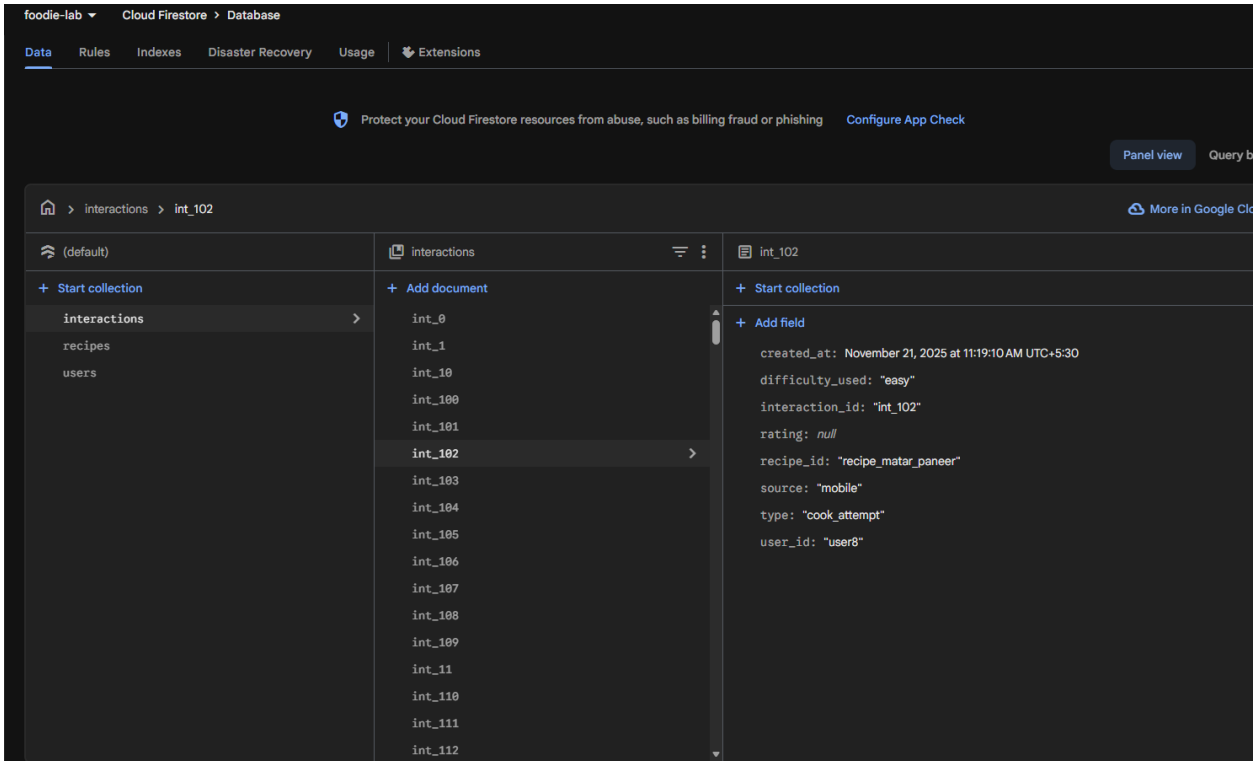
Component	Technology
Database	Firebase Firestore
Programming Language	Python 3.8+
Data Processing	Pandas, NumPy
Visualization	Matplotlib
Firebase SDK	firebase-admin

1.3 System Architecture

2. Data Model Design



2.1 Database Architecture



The system uses Firebase Firestore with a hierarchical document structure optimized for NoSQL patterns.

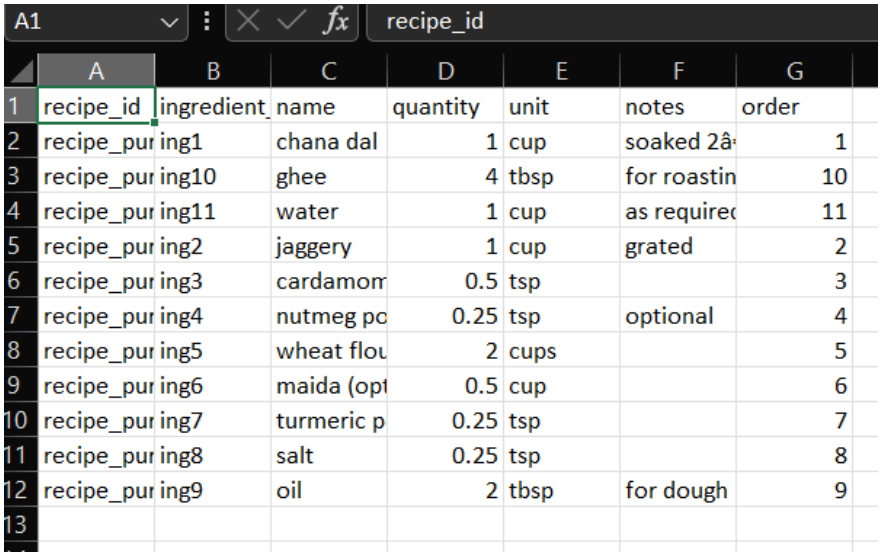
2.2 Design Decisions

Design Choice	Reasoning
Interactions under Recipes	Groups recipe activity together; enables fast queries for single recipe analytics
Activities under Users	Tracks user behavior across recipes; enables user-centric analytics
Denormalized author names	Avoids extra reads; Firestore doesn't support JOINs

2.3 Output Schema

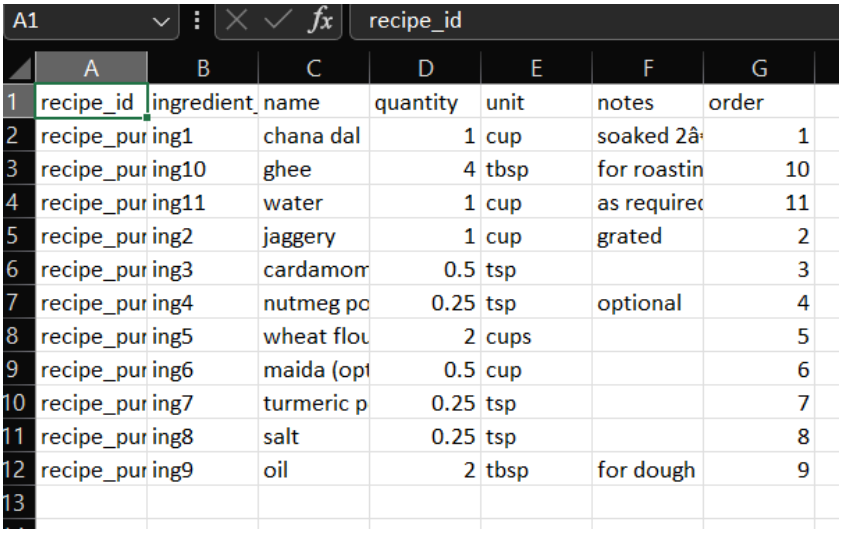
The pipeline produces four normalized CSV tables:

recipes.csv - Primary recipe information including title, description, timing, difficulty, category, dietary type, and author details.



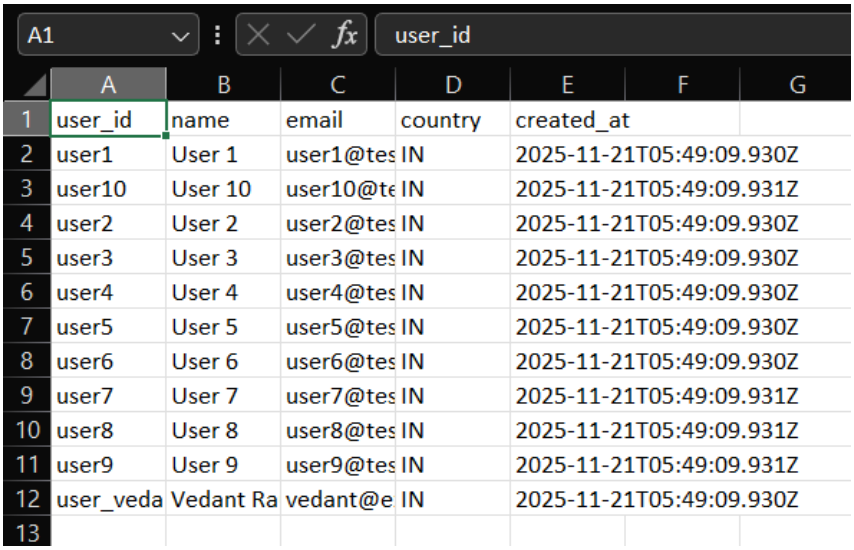
	A	B	C	D	E	F	G
1	recipe_id	ingredient_	name	quantity	unit	notes	order
2	recipe_pur	ing1	chana dal	1	cup	soaked 2â	1
3	recipe_pur	ing10	ghee	4	tbsp	for roastin	10
4	recipe_pur	ing11	water	1	cup	as requirec	11
5	recipe_pur	ing2	jaggery	1	cup	grated	2
6	recipe_pur	ing3	cardamom	0.5	tsp		3
7	recipe_pur	ing4	nutmeg po	0.25	tsp	optional	4
8	recipe_pur	ing5	wheat flou	2	cups		5
9	recipe_pur	ing6	maida (opt	0.5	cup		6
10	recipe_pur	ing7	turmeric p	0.25	tsp		7
11	recipe_pur	ing8	salt	0.25	tsp		8
12	recipe_pur	ing9	oil	2	tbsp	for dough	9
13							

ingredients.csv - Normalized ingredient data linked to recipes via foreign key, containing name, quantity, unit, and optional flag.



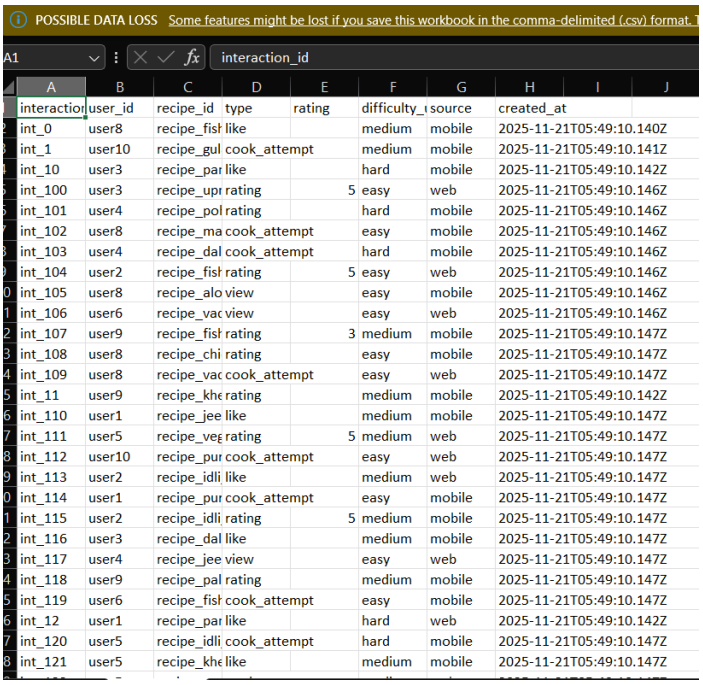
	A	B	C	D	E	F	G
1	recipe_id	ingredient_	name	quantity	unit	notes	order
2	recipe_pur	ing1	chana dal	1	cup	soaked 2â	1
3	recipe_pur	ing10	ghee	4	tbsp	for roastin	10
4	recipe_pur	ing11	water	1	cup	as requirec	11
5	recipe_pur	ing2	jaggery	1	cup	grated	2
6	recipe_pur	ing3	cardamom	0.5	tsp		3
7	recipe_pur	ing4	nutmeg po	0.25	tsp	optional	4
8	recipe_pur	ing5	wheat flou	2	cups		5
9	recipe_pur	ing6	maida (opt	0.5	cup		6
10	recipe_pur	ing7	turmeric p	0.25	tsp		7
11	recipe_pur	ing8	salt	0.25	tsp		8
12	recipe_pur	ing9	oil	2	tbsp	for dough	9
13							

users.csv - Recipe instructions with step numbers, instructions, and duration, linked to parent .



	A	B	C	D	E	F	G
1	user_id	name	email	country	created_at		
2	user1	User 1	user1@tes	IN	2025-11-21T05:49:09.930Z		
3	user10	User 10	user10@te	IN	2025-11-21T05:49:09.931Z		
4	user2	User 2	user2@tes	IN	2025-11-21T05:49:09.930Z		
5	user3	User 3	user3@tes	IN	2025-11-21T05:49:09.930Z		
6	user4	User 4	user4@tes	IN	2025-11-21T05:49:09.930Z		
7	user5	User 5	user5@tes	IN	2025-11-21T05:49:09.930Z		
8	user6	User 6	user6@tes	IN	2025-11-21T05:49:09.930Z		
9	user7	User 7	user7@tes	IN	2025-11-21T05:49:09.931Z		
10	user8	User 8	user8@tes	IN	2025-11-21T05:49:09.931Z		
11	user9	User 9	user9@tes	IN	2025-11-21T05:49:09.931Z		
12	user_veda	Vedant Ra	vedant@e	IN	2025-11-21T05:49:09.930Z		
13							

interactions.csv - User engagement data including ratings, cook notes, and timestamps.



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format.

	A	B	C	D	E	F	G	H	I	J
1	interactio	user_id	recipe_id	type	rating	difficulty_	source	created_at		
2	int_0	user8	recipe_fis	like		medium	mobile	2025-11-21T05:49:10.140Z		
3	int_1	user10	recipe_gul	cook_attempt		medium	mobile	2025-11-21T05:49:10.141Z		
4	int_10	user3	recipe_par	like		hard	mobile	2025-11-21T05:49:10.142Z		
5	int_100	user3	recipe_upı	rating	5	easy	web	2025-11-21T05:49:10.146Z		
6	int_101	user4	recipe_pol	rating		hard	mobile	2025-11-21T05:49:10.146Z		
7	int_102	user8	recipe_ma	cook_attempt		easy	mobile	2025-11-21T05:49:10.146Z		
8	int_103	user4	recipe_dal	cook_attempt		hard	mobile	2025-11-21T05:49:10.146Z		
9	int_104	user2	recipe_fis	rating	5	easy	web	2025-11-21T05:49:10.146Z		
10	int_105	user8	recipe_alo	view		easy	mobile	2025-11-21T05:49:10.146Z		
11	int_106	user6	recipe_vac	view		easy	web	2025-11-21T05:49:10.146Z		
12	int_107	user9	recipe_fis	rating	3	medium	mobile	2025-11-21T05:49:10.147Z		
13	int_108	user8	recipe_chi	rating		easy	mobile	2025-11-21T05:49:10.147Z		
14	int_109	user8	recipe_vac	cook_attempt		easy	web	2025-11-21T05:49:10.147Z		
15	int_11	user9	recipe_khe	rating		medium	mobile	2025-11-21T05:49:10.142Z		
16	int_110	user1	recipe_jee	like		medium	mobile	2025-11-21T05:49:10.147Z		
17	int_111	user5	recipe_veg	rating	5	medium	web	2025-11-21T05:49:10.147Z		
18	int_112	user10	recipe_pur	cook_attempt		easy	web	2025-11-21T05:49:10.147Z		
19	int_113	user2	recipe_idli	like		medium	web	2025-11-21T05:49:10.147Z		
20	int_114	user1	recipe_pur	cook_attempt		easy	mobile	2025-11-21T05:49:10.147Z		
21	int_115	user2	recipe_idli	rating	5	medium	mobile	2025-11-21T05:49:10.147Z		
22	int_116	user3	recipe_dal	like		medium	mobile	2025-11-21T05:49:10.147Z		
23	int_117	user4	recipe_jee	view		easy	web	2025-11-21T05:49:10.147Z		
24	int_118	user9	recipe_pal	rating		medium	mobile	2025-11-21T05:49:10.147Z		
25	int_119	user6	recipe_fis	cook_attempt		easy	mobile	2025-11-21T05:49:10.147Z		
26	int_12	user1	recipe_par	like		hard	web	2025-11-21T05:49:10.142Z		
27	int_120	user5	recipe_idli	cook_attempt		hard	mobile	2025-11-21T05:49:10.147Z		
28	int_121	user5	recipe_khe	like		medium	mobile	2025-11-21T05:49:10.147Z		

3. ETL Process

3.1 Pipeline Architecture

The pipeline follows a four-stage process: Extract → Transform → Validate → Analyze

3.2 Extract Phase

The extraction phase connects to Firebase Firestore using the Admin SDK with streaming for efficient data retrieval. Key operations include authentication via service account credentials, document streaming for large collections, subcollection extraction for interactions, and timestamp conversion to ISO 8601 format.

3.3 Transform Phase

Transformation	Description
Flatten Ingredients	Nested array converted to separate CSV with recipe_id FK
Flatten Steps	Nested array converted to separate CSV with recipe_id FK
Extract Subcollections	Firestore subcollection to interactions.csv
Normalize Time	Structured time object to individual minute columns
Handle Missing Data	Default values: "Uncategorized", "Unknown"

---

4. Data Validation

4.1 Validation Rules

Rule	Field	Criteria
Required Fields	title	Must not be empty
Valid Difficulty	difficulty	Must be: Easy, Medium, Hard, Expert

Rule	Field	Criteria
Prep Time	prep_time_min	Must be > 0
Cook Time	cook_time_min	Must be ≥ 0
Time Logic	total_time_min	Must be ≥ prep_time + cook_time
Ingredient Quantity	quantity	Must be > 0 if numeric
Rating Range	rating	Must be between 0 and 5
Has Steps	steps	At least one step required
Has Ingredients	ingredients	At least one ingredient required

4.2 Validation Output

The validator produces a JSON report containing total recipe count, valid/invalid counts, detailed error messages for invalid records, and list of valid record IDs.

---

5. Analytics & Insights

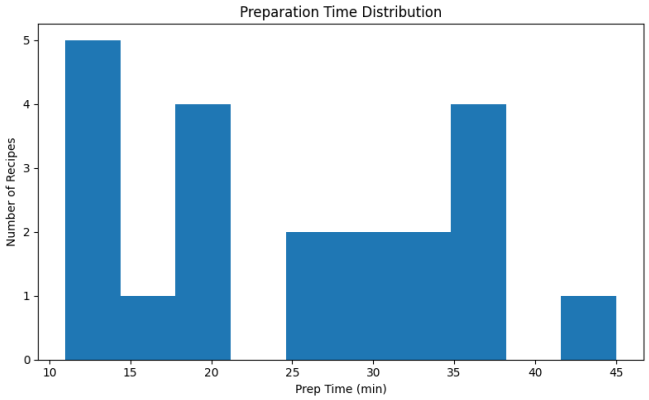
5.1 Generated Insights

The pipeline produces 11 analytical insights:

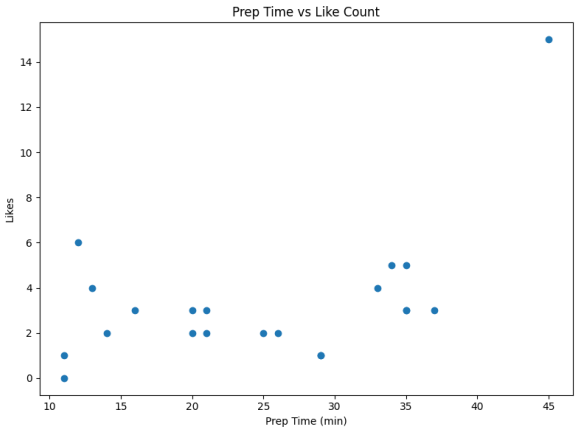
1. **Most Common Ingredients** - Top 20 ingredients by frequency
2. **Average Prep Time** - Mean preparation time in minutes
3. **Average Cook Time** - Mean cooking time in minutes
4. **Difficulty Distribution** - Recipe count per difficulty level
5. **Most Interacted Recipes** - Top 20 by interaction count
6. **Prep vs Rating Correlation** - Statistical correlation analysis

7. **High-Rating Ingredients** - Ingredients appearing in 4+ star recipes
8. **Top Rated Recipes** - Top 10 by average rating
9. **Steps Distribution** - Statistical summary of recipe complexity
10. **Most Commented Recipes** - Top 10 by cook note count
11. **Longest Recipes** - Top 10 by total preparation time

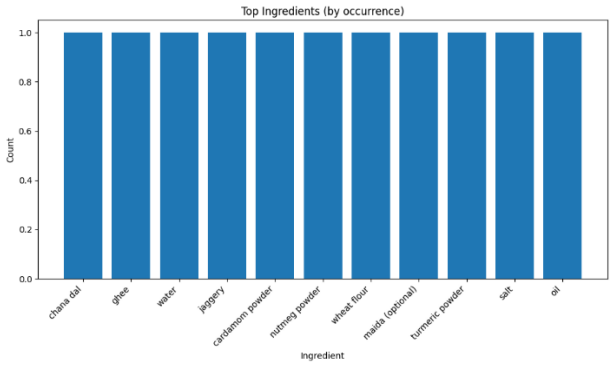
5.2 Visualizations



6.1 Preparation Time Distribution

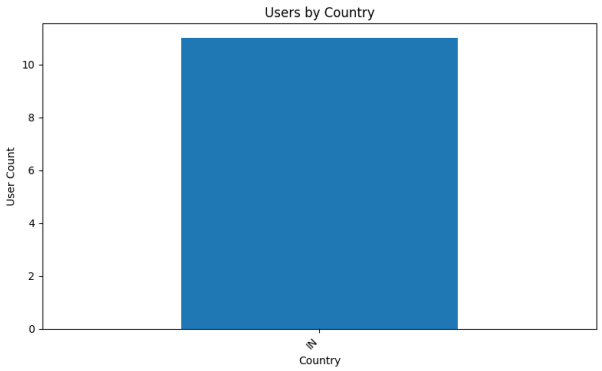


6.2 Prep Time vs Like Count

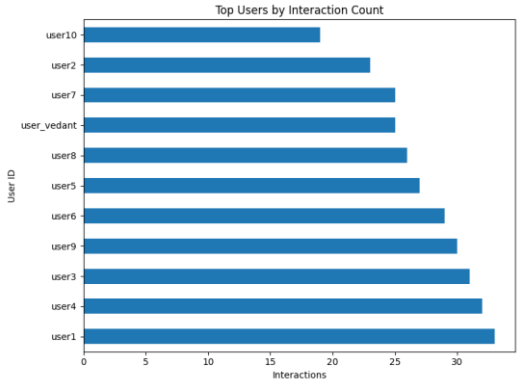


6.3 Top Ingredients (by frequency)

7. User Analytics (Charts)



7.1 Users by Country



7.2 Top Users by Interaction Count

6. Constraints & Limitations

6.1 Firestore Limitations

Constraint	Impact	Mitigation
No native JOINs	Cannot query across collections	Denormalized data; subcollections
Read costs	Each document read is billed	Streaming instead of batch reads
No aggregations	No COUNT/SUM/AVG in queries	Aggregations in Python post-export
Subcollection queries	Cannot query all at once	Iterate per parent document

6.2 Pipeline Constraints

- Sequential execution required (seed → transform → validate → analyze)
- Full export only; no incremental/delta processing
- Memory-bound with pandas DataFrames
- Hardcoded relative paths in some scripts

6.3 Scalability Notes

- Current tested capacity: ~20 recipes, ~200 interactions
- For 1000+ recipes: implement pagination in export
- For larger datasets: consider chunked processing or Apache Spark

---

7. Project Structure

```
RECIPE_ANALYTICS/
├── analytics/          # Analytics outputs and charts
├── config/             # Firebase credentials
├── data_validation/    # Validation scripts and reports
├── Firebase_Setup/     # Data seeding scripts
├── transform_data/     # ETL outputs (CSV files)
└── README.md
```

---

8. Installation & Execution

8.1 Prerequisites

- Python 3.8 or higher
- Firebase project with Firestore enabled
- Service account credentials (JSON)

8.2 Execution Steps

1. **Seed Initial Data** - Run seed_data.py to create base recipe and user
2. **Generate Synthetic Data** - Run genrate_sytetic.py for 20 Maharashtrian recipes
3. **Transform Data** - Run transform.py to export to CSV

4. **Validate Data** - Run validator.py for quality checks
5. **Generate Analytics** - Run analytics.py for insights and charts

---

9. Deliverables Summary

Deliverable	**Status**
Source files for ETL scripts	Complete
Validation script	Complete
Normalized CSV output	Complete
Analytics summary (JSON)	Complete
Documentation	Complete
Visualization charts	Complete

---

10. Conclusion

The Firebase-Based Recipe Analytics Pipeline successfully demonstrates a complete ETL workflow for NoSQL data, producing normalized relational outputs suitable for further analysis. The system handles the unique challenges of Firestore's document model while maintaining data quality through comprehensive validation.
