Customer Complaint Trend Forecasting & Risk Detection Using NLP & Predictive Analysis

1. Introduction

This project focuses on analysing and forecasting consumer complaint volumes across different products and U.S. states, with the aim of identifying which combinations are likely to experience an increase in complaints in the future. Complaints are not only counted but also interpreted through root-cause topic extraction, enabling both quantitative and qualitative insights.

The final output is a consolidated insight system combining forecasting, root-cause attribution, risk scoring, and interactive dashboards.

The project addresses real-world challenges such as incomplete monthly data, duplicated root-cause rows, unstable forecasting behaviour, and inconsistent metadata. A carefully structured pipeline was developed to clean the dataset, generate stable forecasts, and derive interpretable insights.

2. End-to-End Project Pipeline (Structured Overview)

1. Raw Data Ingestion

2. Data Cleaning & Canonicalization

3. Monthly Aggregation & Time Index Continuity Fixing

4. Root-Cause Topic Extraction (NLP)

5. Forecasting with SARIMA (Primary) and Prophet (Comparison)

6. Forecast Quality Filtering & Stable Pair Identification

7. Master BI Dataset Construction

8. Timeline Reshaping for Visualizations

9. Model Evaluation & Metrics Aggregation

10. Insightful Visualization Development (Python + Tableau)

11. Final Documentation & Deliverables

3. Detailed Explanation of Each Pipeline Phase

3.1 Raw Data Ingestion

Complaint data was initially stored across multiple CSV files containing complaint counts, text-based root causes, product classifications, and geographic identifiers.

The dataset was huge and data ingestion was performed using Pyspark, due to its efficiency in loading large datasets and flexibility in performing tabular transformations.

The focus of this stage was to consolidate disparate raw files into a structured environment suitable for pipeline processing

Technologies used:

Python. Pyspark, Pandas, Google Colab file system

3.2 Data Cleaning & Canonicalization

The raw complaint data contained several inconsistencies:

- Irregular product names
- Mixed-case state abbreviations
- Missing or malformed date fields
- Non-standard text patterns in root cause columns
- Duplicate complaint lines
- Null complaint counts in particular months

Cleaning ensured that every Product × State × Month record adhered to strict formatting. Name canonicalization ensured that all product names followed a unified naming convention.

State validation eliminated invalid geographic entries. Date fields were converted to proper datetime format. This stage established a stable foundation for forecasting and merging operations later in the pipeline.

Technologies used:

Pandas (string processing, type casting, deduplication), Python datetime utilities

3.3 Monthly Aggregation & Time Index Continuity Fixing

Complaint records were aggregated to monthly counts for each Product-State pair.

However, many product-state combinations lacked months in the timeline, causing forecasting models like SARIMA and Prophet to behave unpredictably or fail entirely.

A continuity correction algorithm was created:

- Identify missing year-month entries
- Auto-generate missing rows
- Fill them with zero complaints to maintain realistic temporal structure

The corrected dataset produced uniform 60-month sequences for stable forecasting.

Technologies used: Pandas (groupby, reindexing, date_range generation)

3.4 Root-Cause Topic Extraction (NLP)

Complaint narratives were processed and clustered into topics using a combination of keyword extraction and topic modeling.

Each complaint was assigned:

- A topic ID
- Topic keywords representing dominant words
- A human-interpretable label, such as Card-related dispute, Credit reporting error, or Loan servicing issue
- A topic count, summarizing how frequently a complaint matched that topic

This stage enabled the system to connect forecasted complaint spikes to meaningful root causes.

Technologies used: NLTK, Scikit-learn (TF-IDF, topic modeling), Pandas text utilities, KMeans clusering

3.5 Forecasting with SARIMA & Prophet Comparison

Forecasting was initially attempted using both:

SARIMA (from Statsmodels)

FB Prophet

Prophet displayed instability on many pairs due to:

- Short time series
- Abrupt complaint patterns
- Month gaps (before cleaning)

SARIMA, after hyperparameter tuning, consistently delivered smoother, realistic trends across stable monthly timelines.

Forecasts were generated only for product-state pairs with at least 36 months of complete data. Forecasts covered a 12-month future horizon and were later used in risk scoring.

Technologies used: Statsmodels (SARIMA), Prophet (for comparison),Pandas

3.6 Forecast Quality Filtering & Stable Pair Identification

Not all forecasted pairs were reliable. To ensure insight quality, each forecast was evaluated using:

- MAPE (percentage error)
- MAE (absolute error)
- MSE (squared error)
- Structural checks for unrealistic spikes or sudden collapses
- Duration of historical training data
- Pairs failing stability tests were removed.

Finally, 111 stable SARIMA pairs remained. These were used for risk detection and dashboard visualizations.

Technologies used: Scikit-learn metrics, Pandas ranking and filtering logic

3.7 Master BI Dataset Construction

A consolidated dataset was built by merging the following components:

- Historical complaint counts
- Forecasted complaint counts
- Product metadata
- State metadata
- Topic interpretation and keywords
- Model metadata
- Year and month indicators

This file formed the backbone dataset for all BI dashboards and evaluations.

Technologies used: Pandas (merge, join, concatenation)

3.8 Timeline Reshaping for Visualizations

For cross-platform compatibility (Python charts, Tableau dashboards, and Power BI), the master dataset was transformed into a long-form format containing:

- ds (date)
- Product
- State
- actual_count
- forecast_count
- type (Actual or Forecast)

This format ensures that Unified line charts without broken timelines, Easy filtering by product or state and correct alignment of actual vs forecast curves

Technologies used: Pandas (melt, concatenation, tagging)

3.9 Model Evaluation & Metrics Aggregation

All stable SARIMA forecasts were evaluated, and overall model performance was summarized:

- ✓ Average MAPE: 1.46% (excellent relative accuracy)
- ✓ Average MAE: 18.9 complaints
- ✓ Average MSE: 7,846

Interpretation:

Low MAPE indicates consistent forecasting quality across pairs. MAE around 19 complaints is reasonable considering some states have higher complaint volumes and high MSE is expected due to squared-error sensitivity to larger states. SARIMA was therefore selected as the final modelling technique.

Technologies used: Pandas, Scikit-learn, SARIMA

3.10 Insight Visualization (Python + Tableau)

A variety of insights were constructed:

- Trend Line Charts: Shows how complaints evolve over time, contrasting actual and predicted values.
- Top 10 Rising Product-State Segments (Risk Ranking): Ranks combinations expected to experience the highest growth in complaints.
- Top 10 Falling or Stable Segments:Highlights product-state pairs where complaint volumes are declining.
- Root Cause Contribution Charts: Shows dominant complaint topics driving high-risk segments.
- State-Level Complaint Map: Displays aggregated historical complaints geographically.

Each visualization helps contextualize the forecast and supports decision making.

Technologies used: Matplotlib, Seaborn, Tableau, Pandas

4. Conclusion

This project successfully transformed raw consumer-complaint records into a structured forecasting and insight pipeline capable of identifying emerging risks across product–state segments. By combining rigorous data cleaning, time-series modeling with SARIMA, and NLP-based root-cause interpretation, the system provides a reliable view of both historical complaint behavior and future trends. The resulting insights—risky segments, stable segments, dominant complaint causes, and state-level patterns—equip stakeholders with actionable intelligence to prioritize interventions, allocate resources, and mitigate potential surges in consumer dissatisfaction.