Universidad Distrital Francisco José de Caldas

Systems analysis and desing GR020-85

Workshop 1
Competition: Web Traffic Time Series Forecasting

Members:

Johan Sebastián Beltrán Merchán code: 20222020019
Edison David Álvarez Varela code: 20222020043
Yader Ibraldo Quiroga Torres code: 20222020034
Julián David Celis Giraldo code: 20222020041

Lecturer:

Carlos Andrés Sierra Virgüez

Bogotá, Colombia
September 27, 2025

1 Competition overview

The competition chosen for this analysis revolves around predicting the future web traffic of approximately 145,000 Wikipedia articles. For this purpose, participants are provided with historical data of daily page views for each article, which will be used to forecast the number of visits in future dates. Submissions for the contest are evaluated based on an error metric that compares the predicted values against the actual observed values, allowing the accuracy of the forecasting models to be assessed.

The Symmetric Mean Absolute Percentage Error (SMAPE) is a metric used to measure the accuracy of forecasts. This score is calculated as the average of the absolute difference between the predicted and actual values, divided by the mean of their absolute values. It is defined as:

$$SMAP = \frac{100}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2} \tag{1}$$

Where A_t represents the actual value at time t, F_t represents the forecasted value at time t, and n is the number of predictions. By design, SMAPE ranges from 0% to 200%, with 0% indicating a perfect forecast. In the case where both the actual and forecasted values are zero, SMAPE is defined as 0 to avoid division by zero.
In this case, a value of 0% in the evaluation metric represents a perfect prediction, meaning that the forecasted values match exactly with the actual observed values.

The competition data originates from the daily page view logs of approximately 145,000 Wikipedia articles. The primary training file contains this historical traffic, spanning from July 2015 to September 2017. Crucially, the article name itself includes embedded metadata (like language, access type, and agent) which is vital for modeling. This historical series is used to forecast future traffic, and the required submission is managed via key files that map the specific date and article to a unique Id.

- train_*.csv holds all the historical daily traffic where rows are individual articles and columns are dates. The page name is a critical metadata field, encoding details like the Wikipedia project, access type (desktop/mobile), and traffic agent. The data also includes some missing entries.
- key_*.csv is a key map that links the full article page names to the condensed Id used to submit your final predictions.
- sample_submision_*.csv contains the format needed.

The dataset comes with notable restrictions that affect its use. First, it is constrained to fixed time ranges, which limits both the training and evaluation windows available for modeling. Second, the data source does not differentiate between actual zero traffic and missing values, creating ambiguity when interpreting daily page views. Additionally, page-date combinations are represented by shortened Ids to reduce file size, meaning external key files are required to correctly map them back to article names. These restrictions introduce challenges in data preprocessing, interpretation, and accurate forecasting.

2 systemic analysis report

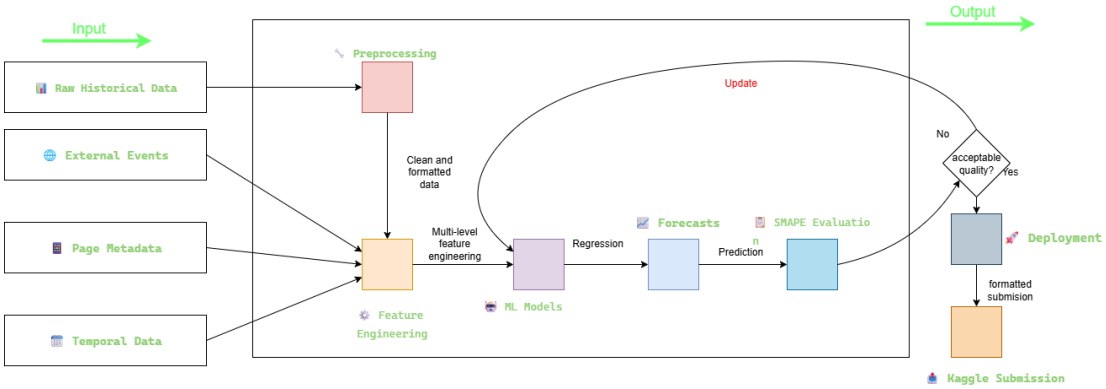

Figure 1: Data flow and their transform.

2.1 Systemic analysis

input components

- **Historical Traffic:** 145K pages × 800+ days of page views.
- **Page Metadata:** Language, project, access method, agent type.
- **External Events:** Holidays, Wikipedia maintenance, viral events.
- **Temporal Data:** Calendar features, seasonal patterns.

Relationships

- All inputs feed into centralized data storage.
- Metadata links page identifiers to traffic records.
- External events correlate with traffic anomalies.
- Calendar data provides temporal context.

Data Characteristics

- **Scale:** Large volume of data (145,000 time series with over 800 time points each).
- **Sparsity:** Many pages have sporadic traffic, with long periods of zero or minimal views.
- **Heterogeneity:** Traffic patterns vary widely across different types of pages (e.g., music, technology, movies).

- **Missing Values:** Some days may have missing data, which could be due to various reasons (e.g., no traffic, data collection issues).

Process

Feature Engineering

Characteristics

- **Temporal Features:** Day of week, month, holidays, seasonal patterns.
- **Page-based Features:** Category, language, project type patterns.
- **Statistical Features:** Rolling averages, volatility measures, trend components.

Specific Problems

- **Feature Proliferation:** Risk of creating too many features (curse of dimensionality).
- **Temporal Leakage:** Avoiding future information in feature creation.
- **Sparse Page Handling:** Features for low-traffic pages are less reliable.
- **Computational Cost:** Calculating rolling statistics for 145K series is expensive.

Model Training & Forecasting

Characteristics

- **Algorithm Diversity:** ARIMA, Prophet, LSTM, ensemble methods.
- **Hierarchical Approach:** Different models for different page types.
- **Validation Strategy:** Temporal cross-validation to prevent data leakage.

Specific Problems

- **Model Selection:** No single algorithm works well for all page types.
- **Computational Constraints:** Training deep learning models on 145K series.
- **Intermittent Demand:** Traditional models struggle with sparse time series.
- **Overfitting Risk:** Complex models may memorize noise rather than patterns.

Validation & Quality Control

Characteristics

- **SMAPE Evaluation:** Symmetric Mean Absolute Percentage Error.
- **Temporal Validation:** Train on past, validate on recent data.
- **Error Analysis:** Identifying systematic prediction errors.

Specific Problems

- **SMAPE Limitations:** Poor handling of zero values and small denominators.
- **Validation Data Sparsity:** Limited recent data for robust validation.
- **Category-wise Performance:** Models may work well for popular pages but fail on niche content.
- **Threshold Setting:** Determining acceptable SMAPE levels for different page types.

Outputs

60-Day Forecasts

Characteristics

- **Time Horizon:** Daily predictions for September 11 – November 10, 2017.
- **Format Requirements:** Integer page views, non-negative values.
- **Uncertainty Quantification:** Prediction intervals for confidence assessment.

Specific Problems

- **Integer Constraint:** Models produce continuous values that must be rounded.
- **Zero Prediction:** Difficult to predict when sparse pages will have traffic.
- **Volatility Capture:** Forecasting both stable trends and viral spikes accurately.
- **Multi-scale Patterns:** Balancing daily, weekly, and monthly seasonality.

SMAPE Evaluation Results

Characteristics

- **Accuracy Metric:** Symmetric Mean Absolute Percentage Error.
- **Benchmarking:** Comparison against naive forecasting methods.
- **Error Distribution:** Analysis of which page types are hardest to predict.

Specific Problems

- **Metric Interpretation:** SMAPE values can be misleading for low-traffic pages.
- **Benchmark Selection:** Choosing appropriate baseline models for comparison.
- **Error Analysis:** Identifying systematic vs. random prediction errors.
- **Performance Variance:** Different accuracy across page categories and traffic levels.

Competition Submission (submission.csv)

Characteristics

- **Format Compliance:** Strict CSV structure required by Kaggle.
- **Page Ordering:** Must match the provided key file exactly.
- **Data Types:** Integer values only, no missing predictions.

Specific Problems

- **Submission Size:** 145,000 rows $\times$ 60 columns = 8.7 million predictions to format.
- **Validation Requirements:** Ensuring no page is missing or duplicated.
- **Integer Rounding:** Handling edge cases in float-to-integer conversion.
- **File Size Optimization:** Managing large CSV files for upload and processing.

2.2 Complexity and sensitivity

The system's complexity arises from the need to process highly detailed and heterogeneous input data, including article IDs, visitor domains, timestamps, and thematic metadata. Handling missing or incomplete records, as well as integrating multiple sources, forces the system to manage numerous internal connections and preprocessing steps, increasing computational and structural complexity. External factors, such as viral news, sudden traffic spikes, or mass access from automated web scrapers, introduce additional challenges, as the model must distinguish between genuine user behavior and anomalous patterns to maintain reliable predictions.
This sensitivity to external and internal variations means that small changes in data quality or unusual access patterns can disproportionately affect outcomes. While low-impact factors, like occasional visits from rarely used domains, may minimally influence predictions, the system remains highly sensitive to sudden surges, bot activity, or inconsistencies in the dataset.

Accommodating these factors within the prediction pipeline adds layers of complexity and requires careful design of preprocessing, normalization, and anomaly-handling mechanisms.
Automated agents, such as spiders and web crawlers, can significantly affect both the complexity and sensitivity of the traffic prediction system. These agents generate high volumes of non-human traffic that can distort visit counts, create artificial spikes, and produce irregular patterns that the model might interpret as genuine user behavior. Handling such activity requires additional preprocessing steps, filtering mechanisms, and anomaly detection, which increase the internal complexity of the system. Moreover, sudden surges caused by crawlers can make the system highly sensitive to outliers, potentially skewing predictions for specific articles or domains if not properly managed.

2.3 Chaos and randomness

The system is inherently susceptible to chaotic behavior due to the non-linear and highly dynamic nature of web traffic. Sudden news events, viral content, or global occurrences can generate extreme spikes in visits to specific articles, producing patterns that are highly unpredictable and sensitive to initial conditions. Even minor variations in the timing or source of visits can propagate through the model, leading to substantial differences in the predicted outcomes. This chaotic behavior is further amplified by the interconnections between metadata features, such as article category, domain of access, and temporal trends, which can interact in complex, non-intuitive ways, making accurate long-term forecasting extremely challenging.

Randomness in the dataset also contributes significantly to system unpredictability. Incomplete or missing records, inconsistent logging, and sporadic traffic from low-frequency domains introduce stochastic elements that the system must accommodate. While preprocessing and imputation strategies can mitigate some effects, the inherent randomness of human web browsing behavior ensures that the input data retains a level of unpredictability that cannot be fully modeled. This randomness forces the system to constantly adapt, increasing the difficulty of maintaining robust and stable predictions across all articles and temporal scales.

Additionally, the presence of automated agents, such as web spiders and bots, adds another layer of chaotic influence. These agents can generate bursts of traffic that mimic human behavior, yet follow irregular or programmed patterns, further obscuring the distinction between genuine and artificial visits. Their activity can create seemingly recurring patterns that are, in fact, random and externally driven, injecting additional noise into the system. Accounting for this factor requires complex filtering and anomaly detection mechanisms, which themselves can introduce sensitivity to small changes in the dataset, creating feedback loops where minor inconsistencies escalate into significant prediction errors.

Chaotic elements identified

- Butterfly effect: is evident in this system, as small changes in input can trigger disproportionate variations in predictions. These tiny perturbations propagate through the model's complex interactions between article IDs, domains, timestamps, and thematic data, making forecasts highly sensitive to initial conditions and emphasizing the system's inherent unpredictability.
- Strange attractors: appear as recurrent but non-periodic traffic patterns, such as seasonal peaks in educational content or repeated interest in trending topics. While these attractors provide some structure in the otherwise chaotic data, their irregular nature combined with random human behavior and automated agents like spiders introduces additional complexity. The model must capture these subtle patterns without overfitting to noise, balancing order and chaos to maintain reliable predictions across diverse articles.

2.4 Conclusion

Web traffic data through Wikipedia articles can be highly complex. We observe chaotic elements, where a small news item or a temporary trend can generate massive traffic for just a few hours or days, which may affect sensitivity and destabilize the prediction model. Likewise, strange attractors can emerge in the form of recurrent but non-periodic patterns, such as educational traffic during school cycles, as well as similarities in traffic patterns across different temporal scales. The historical data is recorded day by day, which is advantageous for grouping by days, weeks, special dates, and weekends, allowing for the analysis and correlation of different elements.

It is established that Wikipedia, being a consensus-based system, involves both editors and consumers (readers) who regulate the content generated by editors.

Successful Implementation Requires

- Robust handling of sparse and intermittent time series
- Incorporation of external factors and page metadata
- Careful validation strategies respecting temporal dependencies
- Scalable computational approaches for handling 145,000 parallel time series

The competition provides an excellent framework for developing and testing advanced forecasting methodologies in a real-world, large-scale scenario.

System Nature Identified

Key Findings

Strengths of the System

- Robust evaluation framework (Kaggle's SMAPE) providing balanced performance measures

- Abundant temporal structure: 550 days of historical data enabling the identification of complex patterns
- Temporal consistency: Regular daily samples

Identified Weaknesses

- Data quality issues: Missing values and noise impacting model accuracy, such as query data generated by bots
- Temporal limitation: Fixed training history may not fully adapt to real-time conditions