|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Wed Traffic Time series Analysis** | A computer with a screen on  Description automatically generated | | A graph on a screen  Description automatically generated | Website Analytics traffic Graph image - Free stock photo - Public ... | |
| Final Project  IST652 - Scripting for Data Analysis M001: Fall 2023  by  Anjana Sowmya Puvvada  Srushtee pawar |

**Introduction**

During the pandemic, the surge in internet usage has led to a substantial increase in web traffic for various websites. Managing this increased traffic is crucial to prevent website crashes and downtime, ensuring a seamless user experience. To address this challenge, a time series forecasting approach is employed to predict future web traffic, enabling better traffic control decisions.

Web traffic, defined as the number of requests sent and received by users, forms a significant portion of internet traffic. The objective is to use time series analysis to understand and predict patterns in this data. Time series data, organized chronologically, provides a historical perspective on web traffic trends.

The project involves leveraging time series forecasting techniques to make predictions about future web traffic. Such predictions aid in proactive measures like implementing load balancing to distribute traffic efficiently and prevent overloads on servers. The goal is to enhance the website's performance and user experience.

Time series analysis and forecasting are widely used in various domains, including finance, weather forecasting, smart home monitoring, and supply chain management. The relevance of these techniques in the context of web traffic underscores their importance in making informed decisions for optimizing website performance and ensuring user satisfaction.

**Data Source: Kaggle API**

We have utilized the Kaggle API to extract relevant data from the Kaggle platform. The Kaggle API allows for programmatic interactions with Kaggle, and by running a specific command, I downloaded the dataset directly to my local environment. The command typically involves specifying the dataset using the -d flag followed by the username of the dataset author and the dataset name. This streamlined the data acquisition process, enabling me to focus on tasks such as data exploration, cleaning, and time series analysis. The project aims to uncover insights from the web traffic data, including visualizing trends over time and potentially building predictive models for future traffic patterns.

A screenshot of a computer

Description automatically generated

We have taken the username and key token information from the Kaggle ,by creating a new token

A screenshot of a computer program

Description automatically generated

The code proceeds to download the competition data, identified by the competition ID "web-traffic-time-series-forecasting," utilizing the Kaggle CLI. Subsequently, the downloaded data, likely in a compressed format, is unzipped and extracted to the directory specified as "/content."

The training dataset consists of approximately 145k time series. Each of these time series represents a number of daily views of a different Wikipedia article, starting from July, 1st, 2015 up until December 31st, 2016.

The dataset contains traffic data, where each row corresponds to a particular article and each column correspond to a particular date. Some entries are missing data. The page names contain the Wikipedia project (e.g. en.wikipedia.org), type of access (e.g. desktop) and type of agent (e.g. spider). In other words, each article name has the following format: 'name\_project\_access\_agent' (e.g. 'AKB48\_zh.wikipedia.org\_all-access\_spider')

**Data exploration and data cleaning steps**

**Reading Data**

The Pandas library is then employed to read the training dataset, specifically the file named 'train\_1.csv.zip,' and the resulting data is loaded into a Pandas Data Frame named 'train.' Finally, a preview of the dataset is displayed by showcasing the first few rows using the head() method, offering an initial insight into the structure and content of the data.

Size of data is 145063 rows and 550 columns

A screenshot of a computer

Description automatically generated

**Transforming Data: Rearranging Rows and Columns for Time Series Analysis**

Initially, the rows and columns of the dataset are transposed, treating each row as a separate time series.

The "Page" column is renamed to "Date" for clarity, and its data type is checked and converted to datetime format.

The "Date" column is set as the index, establishing it as the primary axis for time series analysis. The original dataset transformed into a more structured and interpretable format suitable for time series exploration, ensuring proper indexing and data types for temporal analysis.

A computer screen shot of a code

Description automatically generated

Transformed data has 550 rows and 145063 columns.

A screenshot of a computer

Description automatically generated

**A description of data exploration and data cleaning steps**

Checked for null values in the dataset, identifying columns with the most null values.

**A screen shot of a computer program

Description automatically generated**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

We had NaN values. We have analyzed distribution and effect of null values on our analysis for We replaced them with 0 since the dataset does not distinguish between 0 and missing.

Also, It's worth noting that there are non-Wikipedia URLs within the dataset that may not conform to the regex search. These are primarily Wikimedia pages, for which we will assign the code 'na' as their language hasn't been determined. Many of these non-conforming URLs could represent elements like images lacking a clear language association.

**Q1. How Data is distributed based on different types Wikipedia project (e.g. en.wikipedia.org), type of access (e.g. desktop) and type of agent (e.g. spider)**

**A screen shot of a computer

Description automatically generated**

**A pie chart of different types of access

Description automatically generated**

The pie chart shows that the majority of access to the Wikipedia comes from all other access types (such as tablet and tablet web access) (51.2%). Mobile web access accounts for 24.8% of access, and desktop computers account for the remaining 24%.

**Desktop:** This includes access from traditional desktop computers, as well as laptops and other devices that are typically used in a desktop-like setting.

**Mobile web:** This includes access from smartphones and other mobile devices that are using a web browser.

**All other access:** This includes access from devices such as tablets, e-readers, and smart TVs. It also includes access from non-web browsers, such as Wikipedia apps.

**Type of Agent**

**A screenshot of a computer

Description automatically generated**

**A pie chart with text and numbers

Description automatically generated**

The pie chart shows the distribution of access types based on the type of agent. An agent is a computer program that acts on behalf of a user. In the context of the pie chart, agents are used to access the Wikipedia in ways that are not done by humans. For example, search engine crawlers are agents that access the Wikipedia to index its content.

The pie chart shows that the majority of agent access to the Wikipedia comes from Other types of agents, such as Wikipedian bots and spam bots, account for the remaining 75.9%,and the rest comes from spider(search engine crawlers) 24.1%

Other types of agents, such as Wikipedian bots and spam bots, account for the remaining 75.9%.

Here is a breakdown of the agent types:

**Search engine crawlers**: These are bots that are used by search engines to index the content of the Wikipedia.

**Other agents:** This includes bots that are used for tasks such as editing Wikipedia articles, detecting spam, and monitoring website performance.

**Exploring the number of views for different projects**

**A screen shot of a computer program

Description automatically generated**

**A graph of blue bars with black text

Description automatically generated**

The graph shows that the English Wikipedia (en.wikipedia.org) is the most popular Wikipedia project, with over 20 billion page views. It is followed by the Japanese Wikipedia (ja.wikipedia.org) with over 15 billion page views, the German Wikipedia (de.wikipedia.org) with over 10 billion page views, and the French Wikipedia (fr.wikipedia.org) with over 8 billion page views.

The graph also shows that there is a significant long tail of Wikipedia projects with fewer page views. The bottom 20% of Wikipedia projects account for less than 2% of all page views.

**Checking of null value percentage for each project in the data**

**A screenshot of a computer screen

Description automatically generated**

The analysis is also conducted based on different Wikipedia projects, including "zh.wikipedia.org," "fr.wikipedia.org," "en.wikipedia.org," "commons.wikimedia.org," "ru.wikipedia.org," "www.mediawiki.org," "de.wikipedia.org," "ja.wikipedia.org," and "es.wikipedia.org." columns with projects commons.wikimedia.org and www.mediawiki.org have 48% columns with null values. We are unsure if they should be treated as zero visits to a Page on a specific day or maybe something happend on the server end and failed to update numbers. Either way we need to somehow imply that similar patterns can be applied in the future. Since, commons.wikimedia.org and www.mediawiki.org have 48% columns with null values. I would consider other types of projects for my further analysis

* The three Wikipedia projects with the most columns are en.wikipedia.org, zh.wikipedia.org, and commons.wikimedia.org. These are also the three largest Wikipedia projects by number of articles.
* The commons.wikimedia.org project has the highest percentage of nulls. This is likely due to the fact that it is a media repository, and many of the media files do not have all of the metadata fields filled in.
* The ru.wikipedia.org project has the lowest percentage of nulls. This is likely due to the fact that it is a relatively new Wikipedia project, and the editors have been more careful to fill in all of the metadata **fields.**

**Q2- Is Traffic Influenced by Page Language?**

**Unit of Analysis: Language**

**A screen shot of a computer program

Description automatically generated**

**A graph of blue bars

Description automatically generated with medium confidence**

The number of pages in different languages varies widely. English has over 6 million pages, while Russian has less than 1.5 million pages. This is likely due to the different languages and cultures that are represented on Wikipedia.

The top 3 Wikipedia projects by number of pages are all in English (en), Japanese (ja), and Chinese (zh). These are also the 3 largest Wikipedia projects by number of articles.

The Russian Wikipedia project (ru) has the fewest pages of all the Wikipedia projects shown in the graph. It is also the smallest Wikipedia project by number of articles.

The French Wikipedia project (fr) has more pages than the German Wikipedia project (de), but the German Wikipedia project has more articles. This suggests that the German Wikipedia project uses more pages per article than the French Wikipedia project.

**The Time series plot for each language page views**

**A screen shot of a computer program

Description automatically generated**

**A graph of different languages

Description automatically generated**

The graph shows that the English Wikipedia (en.wikipedia.org) is the most popular Wikipedia project in all languages, with over 20 billion page views. It is followed by the Japanese Wikipedia (ja.wikipedia.org) with over 15 billion page views, the German Wikipedia (de.wikipedia.org) with over 10 billion page views, and the French Wikipedia (fr.wikipedia.org) with over 8 billion page views.

More detailed interpretation of some of the key findings from the graph:

The English Wikipedia is the most popular Wikipedia project in all languages. This is likely because the English language is the most widely spoken language in the world.

The number of page views for each Wikipedia project has been increasing over time. This suggests that Wikipedia is becoming increasingly popular in all languages.

**The Highest viewed pages for each language**

**A computer screen shot of a program code

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

This Data output clearly shows the top 5 most viewed pages for each language

**Model Predictions**

**LSTM Model:**

Long Short Term Memory is a type of recurrent neural network capable of learning order dependence in sequence prediction problems.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically the default behavior. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.

A screen shot of a computer program

Description automatically generated

A graph of a diagram

Description automatically generated with medium confidence A graph of a graph

Description automatically generated

First filters the training data using a Fourier transform to remove high-frequency noise. Then,a MinMaxScaler to standardize the data. Next, it trains an LSTM model on the filtered and standardized data. Finally, it uses the trained model to predict future values on the test data.

**A screenshot of a computer program

Description automatically generated**

**A graph with a red and blue line

Description automatically generated**

The plot of the predicted web traffic values (red) and the actual web traffic values (blue). The predicted values seem to follow the general trend of the actual web traffic values, but there are some discrepancies. This is not surprising, as LSTM models are not perfect and can make mistakes.

**ARIMA Model:**

ARIMA stands for AutoRegressive Integrated Moving Average. It is a class of statistical models for analyzing and forecasting time series data.

A quick breakdown of the components of the ARIMA model:

**AR- AutoRegression:** Model that uses the dependent relationship between an observation and some number of lagged observations.

**I- Integrated:** The use of differencing of raw observations in order to make the time series stationary.

**MA- Moving Average:** A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.

ARIMA models describe the trends and seasonality in time series as a function of lagged values and Averages changing over time intervals.

The parameters of the ARIMA model are as follows:

**p** - The number of lag observations included in the model, also called the lag order.

**d**- The number of times that the raw observations are different, also called the degree of differencing.

**q**- The size of the moving average window, also called the order of moving average.

A computer screen shot of a program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

A graph of a number of days

Description automatically generated with medium confidence A graph of data and numbers

Description automatically generated with medium confidence

**A graph of data and numbers

Description automatically generated with medium confidence A graph of data and numbers

Description automatically generated with medium confidence**

**A graph of data with orange and blue lines

Description automatically generated A graph with numbers and lines

Description automatically generated**

**A graph with orange lines

Description automatically generated A graph with blue and orange lines

Description automatically generated**

This predictions show the results of fitting an ARIMA model to the daily web traffic data for the top page in the English language. The plot shows the actual web traffic data (blue line) and the ARIMA model predictions (red line).

The ARIMA model seems to have captured the general trend of the web traffic data, but it is not perfectly accurate. There are some discrepancies between the predicted values and the actual values, especially in the latter part of the time series.

This is not surprising, as ARIMA models are not perfect and can make mistakes. ARIMA models are also sensitive to the order of the model, which is the number of autoregressive (AR), integrated (I), and moving average (MA) terms in the model. In this case, the ARIMA model was fitted with an order of (2, 1, 4), which is a common order for many time series datasets.

Overall, the ARIMA model seems to be a good fit for the web traffic data. It is able to capture the general trend of the data and can be used to make reasonable predictions about future web traffic.

**Further Exploration**

1. Comparing the data with other sources to validate observations and identify additional insights. Investigating the pageviews of specific categories or articles within each language to understand the drivers behind the broader trends.
2. **Seasonality and Trends**:

Analyze the data for recurring seasonal patterns (e.g., weekdays, months) and long-term trends (e.g., yearly growth). Use of models that can capture these patterns and trends, such as ARIMA with seasonal components or LSTMs with seasonal features. This improves the accuracy of predictions by accounting for predictable variations in web traffic.

1. **External Factors:**

Consider additional data sources that might influence web traffic, like holidays, weather, or social media trends. Incorporate these factors into the model as features or use them to adjust predictions dynamically. This provides a more holistic view of web traffic and improves prediction accuracy by accounting for external influences.

1. **Explain ability and Interpretability**:

Use models that explain their predictions, such as decision trees or rule-based models. This helps to understand why the model makes certain predictions and gain insights into the factors affecting web traffic. This knowledge can be used to improve the model and make informed decisions based on the underlying factors.

**Reference**<https://www.kaggle.com/code/ganeshhalpatrao/web-traffic-time-series-forecasting#MODELS>

<https://www.kaggle.com/code/bodamanojkumar/web-traffic-time-series-forecasting-eda#Popular-pages-in-%22en.wikipedia.org%22>

<https://www.kaggle.com/competitions/web-traffic-time-series-forecasting/data>