Business Case Study : Web Page view forecasting using time series analyses

Problem Description

Web page view forecasting is predicting future web traffic based on historical data. This involves analyzing past page views to identify trends, seasonal patterns, and potential anomalies. It helps businesses plan their resource allocation, ensuring that servers and infrastructure can handle expected traffic, thus preventing downtime and improving user experience. Accurate forecasts also enable better marketing strategies by predicting peak times for visitor engagement. Additionally, it aids in revenue estimation and advertising planning, as businesses can anticipate future traffic and adjust their strategies accordingly. Overall, web page view forecasting supports resource allocation, server scaling, and improving user experience by anticipating traffic spikes and ensuring website reliability

The case study aims to understand the per-page view report for different Wikipedia pages for 550 days and forecast the number of views so that the digital ad company can predict and optimize the ad placement for their clients. The dataset consists of 145k Wikipedia pages and the daily view count for each of them. The clients belong to different regions and need data on how their ads will perform on pages in the various languages of the website.

The dataset consists of the two CSV files given below:

1. train_1.csv : In the csv file, each row corresponds to a particular article and each column corresponds to per page view count for the given dates.
   The page name contains data in this format:
   SPECIFIC NAME _ LANGUAGE.wikipedia.org _ACCESS TYPE_ACCESS ORIGIN
   having information about the page name, the main domain, the device type used to access the page, and also the request origin (spider or browser agent)

2. Exog_Campaign_eng: This file contains data for the dates with a campaign or significant event that could affect the views for that day. The data is just for pages in English. There's 1 for dates with campaigns and 0 for remaining dates. It is to be treated as an exogenous variable for models when training and forecasting data for pages in English

Business Questions to be answered from Analysis

1. Defining the problem statements and where can this and modifications of this be used?
2. Write 3 inferences you made from the data visualizations
3. What does the decomposition of series do?
4. What level of differencing gave you a stationary series?
5. Difference between ARIMA, SARIMA & SARIMAX.
6. Compare the number of views in different languages
7. What other methods other than grid search would be suitable to get the model for all languages?

Methodology

There are two sets of files in the given dataset. The first two CSV files have to be analyzed to find the characteristics of the dataset. The steps include

1. **Exploratory Data Analysis**

   Begin by importing the dataset and performing exploratory analysis to understand its structure and characteristics, such as data types and column names. Next, separate the data into relevant subsets for more detailed analysis. Analyze and visualize these subsets to uncover patterns, trends, and relationships. Finally, draw inferences from your analysis to gain insights and inform further steps in your data analysis or modeling process.

2. **Checking stationarity**

To check for stationarity in your time series data, start by formatting the data appropriately for the model. Perform the Dickey-Fuller test to statistically determine if the data is stationary. Use decomposition to break down the time series into trend, seasonal, and residual

components. If the data is not stationary, apply differencing to transform it into a stationary series, making it suitable for time series modeling.

3. **Creating model training and forecasting with ARIMA, SARIMAX**

   To create model training and forecasting with ARIMA and SARIMAX, start by plotting the ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) to identify the order of the models. Train the model using historical data, ensuring it captures the underlying patterns. Perform forecasting for different languages or regions as needed. Finally, plot the forecasted results to visualize the model's predictions and assess its accuracy.

4. **Forecasting with the Facebook prophet**

   Forecasting with Facebook Prophet involves using an open-source tool designed for time series forecasting. Prophet models data based on an additive approach, capturing non-linear trends, seasonality, and holiday effects. It is particularly effective for datasets with strong seasonal patterns and multiple seasons of historical data. The tool is robust to missing data and outliers, making it a versatile choice for accurate and automated forecasts.

5. **Creating a pipeline for working with multiple series**

   Creating a pipeline for working with multiple time series involves designing a systematic process to handle and analyze each series efficiently. This includes steps like data preprocessing (cleaning and formatting), feature engineering, and model training. The pipeline should be capable of handling various series simultaneously, applying consistent transformations and models. It ensures scalability and reproducibility, making it easier to manage and forecast multiple time series datasets effectively.

Analysis

1. **Exploratory Data Analysis**

   The first CSV file "train_1.csv" is imported and its data structure is analyzed. It consists of 145063 rows and 551 columns and its sample data structure is shown in Figure 2.1. The row is in the format "SPECIFIC NAME _ LANGUAGE.wikipedia.org _ACCESS TYPE_ACCESS ORIGIN" having information about the page name, the main domain, the device type used to

access the page, and also the request origin and the columns are the dates in which the data are acquired.

	Page	2015-07-01	2015-07-02	2015-07-03	2015-07-04	2015-07-05	2015-07-06	2015-07-07	2015-07-08	2015-07-09	...	2016-12-22	2016-12-23	2016-12-24	2016-12-25	2016-12-26	2016-12-27	2016-12-28	2016-12-29	2016-12-30	2016-12-31
0	2NE1_zh.wikipedia.org_all-access_spider	18.0	11.0	5.0	13.0	14.0	9.0	9.0	22.0	26.0	...	32.0	63.0	15.0	26.0	14.0	20.0	22.0	19.0	18.0	20.0
1	2PM_zh.wikipedia.org_all-access_spider	11.0	14.0	15.0	18.0	11.0	13.0	22.0	11.0	10.0	...	17.0	42.0	28.0	15.0	9.0	30.0	52.0	45.0	26.0	20.0
2	3C_zh.wikipedia.org_all-access_spider	1.0	0.0	1.0	1.0	0.0	4.0	0.0	3.0	4.0	...	3.0	1.0	1.0	7.0	4.0	4.0	6.0	3.0	4.0	17.0
3	4minute_zh.wikipedia.org_all-access_spider	35.0	13.0	10.0	94.0	4.0	26.0	14.0	9.0	11.0	...	32.0	10.0	26.0	27.0	16.0	11.0	17.0	19.0	10.0	11.0
4	52_Hz_I_Love_You_zh.wikipedia.org_all-access_s...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	48.0	9.0	25.0	13.0	3.0	11.0	27.0	13.0	36.0	10.0

Figure 2.1: Sample data structure of the CSV file “train_1.csv”

The page name column “Page” is split into four columns “Specific_Name”, “Langwiki”, “Access_type”, and “Access_origin” and a fifth column named “Language” is added by extracting the language information from the “Langwiki”. (Map the short form of the language to the corresponding language for example "es" is the short form for Spanish.) The data frame after removing “Page” and “Langwiki” columns is displayed in Figure 2.2.

	Specific_Name	Access_type	Access_origin	Language	2015-07-01	2015-07-02	2015-07-03	2015-07-04	2015-07-05	2015-07-06	...	2016-12-22	2016-12-23	2016-12-24	2016-12-25	2016-12-26	2016-12-27	2016-12-28	2016-12-29	2016-12-30	2016-12-31
0	2NE1	all-access	spider	Chinese	18.0	11.0	5.0	13.0	14.0	9.0	...	32.0	63.0	15.0	26.0	14.0	20.0	22.0	19.0	18.0	20.0
1	2PM	all-access	spider	Chinese	11.0	14.0	15.0	18.0	11.0	13.0	...	17.0	42.0	28.0	15.0	9.0	30.0	52.0	45.0	26.0	20.0
2	3C	all-access	spider	Chinese	1.0	0.0	1.0	1.0	0.0	4.0	...	3.0	1.0	1.0	7.0	4.0	4.0	6.0	3.0	4.0	17.0
3	4minute	all-access	spider	Chinese	35.0	13.0	10.0	94.0	4.0	26.0	...	32.0	10.0	26.0	27.0	16.0	11.0	17.0	19.0	10.0	11.0
4	52_Hz_I_Love_You	all-access	spider	Chinese	NaN	NaN	NaN	NaN	NaN	NaN	...	48.0	9.0	25.0	13.0	3.0	11.0	27.0	13.0	36.0	10.0

Figure 2.2: The modified the data frame

There are three access types of which 65713 are of all-access type and 30923 mobile-web and 30572 desktop type as illustrated in Figure 2.3. There are two Access_origin types: all-agents (96594) and spider (30614) as plotted in Figure 2.4. there are seven language types considered. The count and language names are displayed in Figure 2.5. English and Spanish have the higher median values. Chinese has the lowest median value. “All Access” type has the highest percentage of access type (more than 50 %). “All agents” access origin has the highest percentage of access origin type (above 75 %).

2. Checking stationarity

The data frame is grouped based on the language and transformed and the modified data frame consists of “Language” as columns and date as rows are formed as illustrated in Figure 2.6 (sample), the time series plot is plotted in Figure 2.7. The median view plot of all seven languages is shown in Figure 2.8.

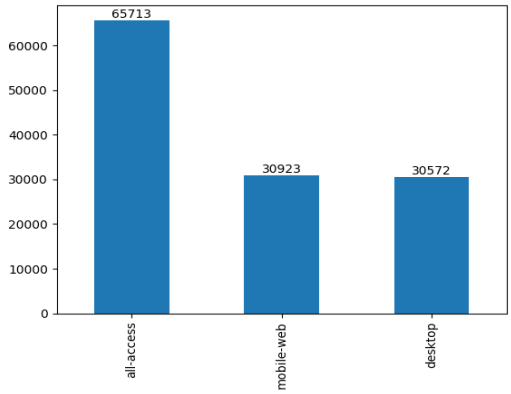


Figure 2.3: Count plot of Access Type

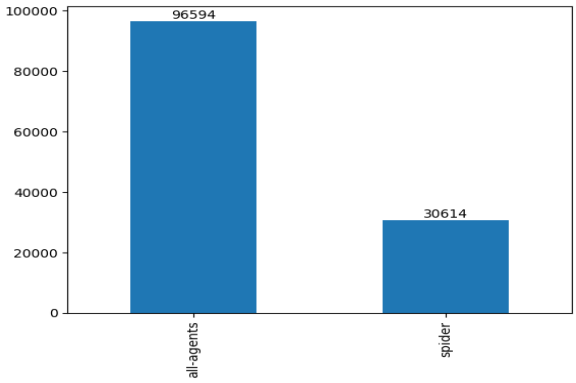


Figure 2.4: Count plot of Access_origin

The median values of speakers for the different languages listed are as follows: Spanish **(**441.25), English (412.50), Russian (311.50), Japanese (233.50), French (147.00), German (138.00), and Chinese (104.00). Using the Dickey fuller test, the stationarity of all the language is checked and found to be non-stationary. The decomposition of all the time series of the language was done. The decomposition sample of the English language is showed Figure 2.9.

All the time series data are differenced and the stationarity of the given time series data was rechecked and found to be stationary. The differencing sample of the English language is plotted in Figure 2.10.

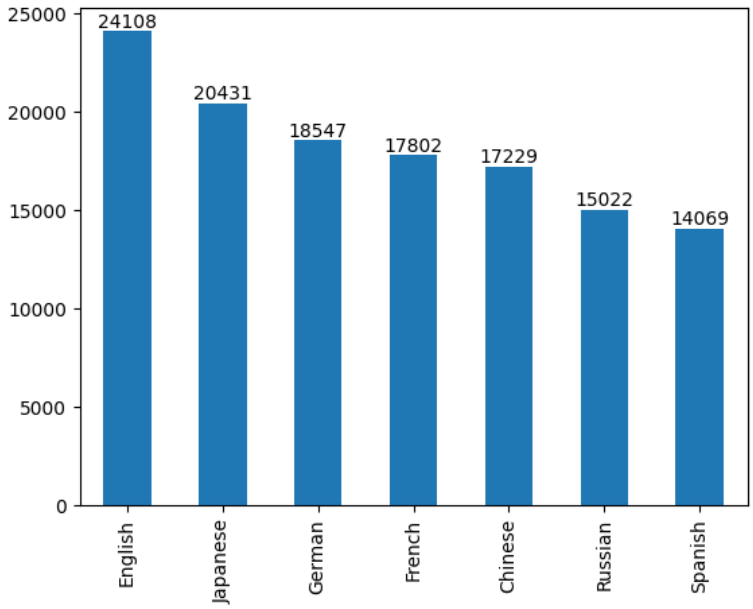


Figure 2.5: Count of Language.

Language	Chinese	English	French	German	Japanese	Russian	Spanish
2015-07-01	72.0	243.0	101.0	99.0	172.0	180.0	368.0
2015-07-02	68.0	242.0	102.0	97.0	167.0	183.0	362.0
2015-07-03	67.0	221.0	100.0	95.0	176.0	182.0	322.0
2015-07-04	69.0	215.0	100.0	96.0	192.0	177.0	261.0
2015-07-05	72.0	232.0	112.0	113.0	190.0	189.0	311.0

Figure 2.6: Data time series sample ready for testing

3. Creating model training and forecasting with ARIMA, SARIMAX

The ACF and PACF plots of all languages are shown in Figures 2.11 and 2.12 respectively. The time series data of the English language is chronologically split into train and tests. Using grid search the parameters of the ARIMA model have been searched. Forecasting for different languages done using the ARIMA model and the corresponding Mean Absolute Error (MAE),

Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are summarized in Table 2.1 and predicted and actual values for the different language are plotted in Figure 2.13. Forecasting for different languages done using the SARIMA model and the corresponding Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are summarized in Table 2.2. Forecasting for different languages done using the SARIMAX model and the corresponding Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are summarized in Table 2.3.

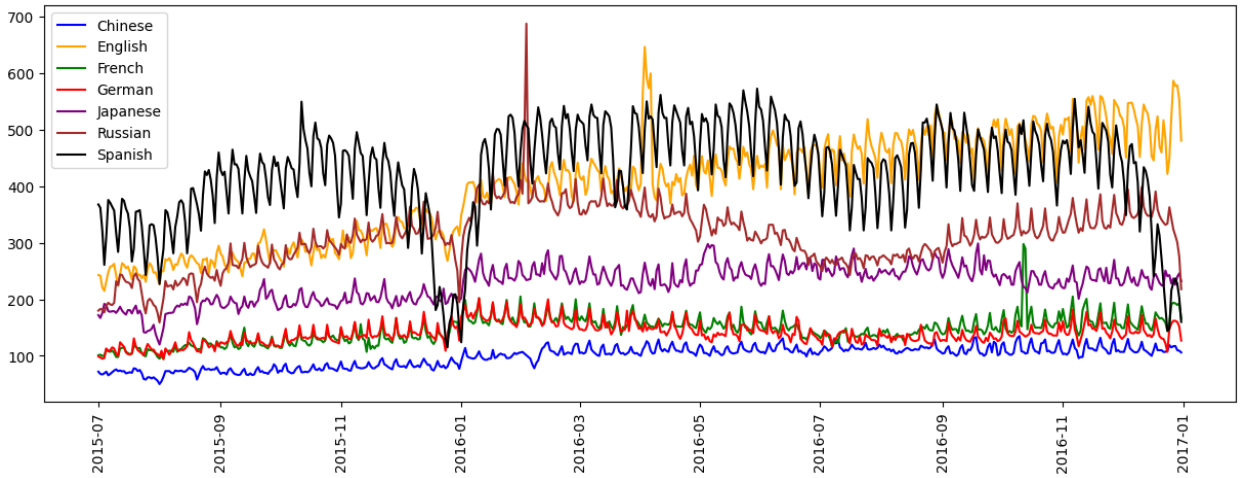


Figure 2.7: Time Series plot of all the seven languages

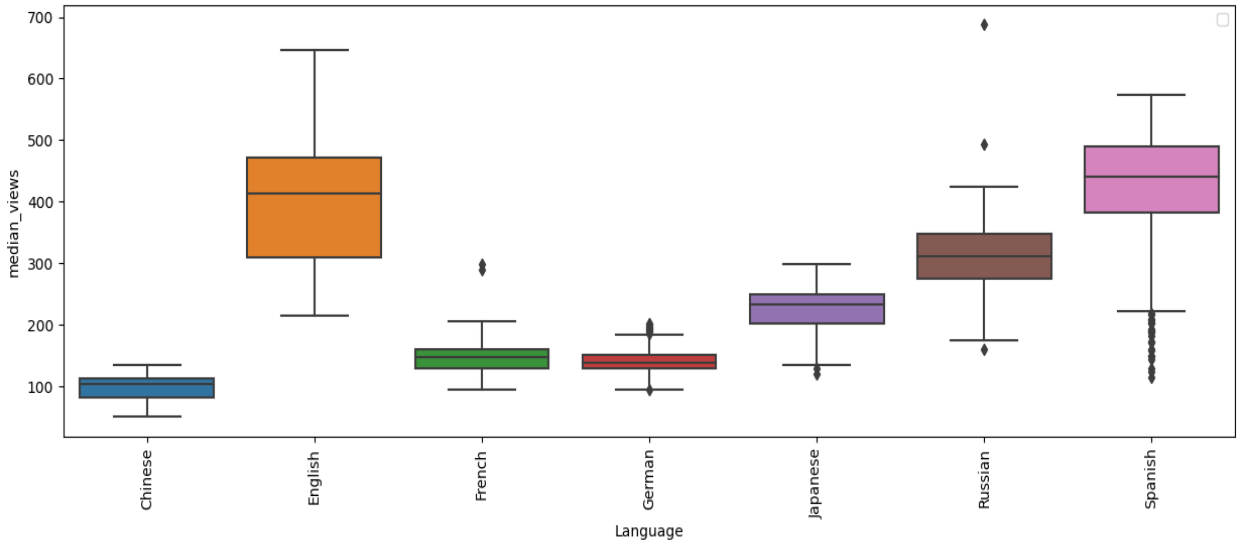


Figure 2.8: Median view plot of all the seven languages

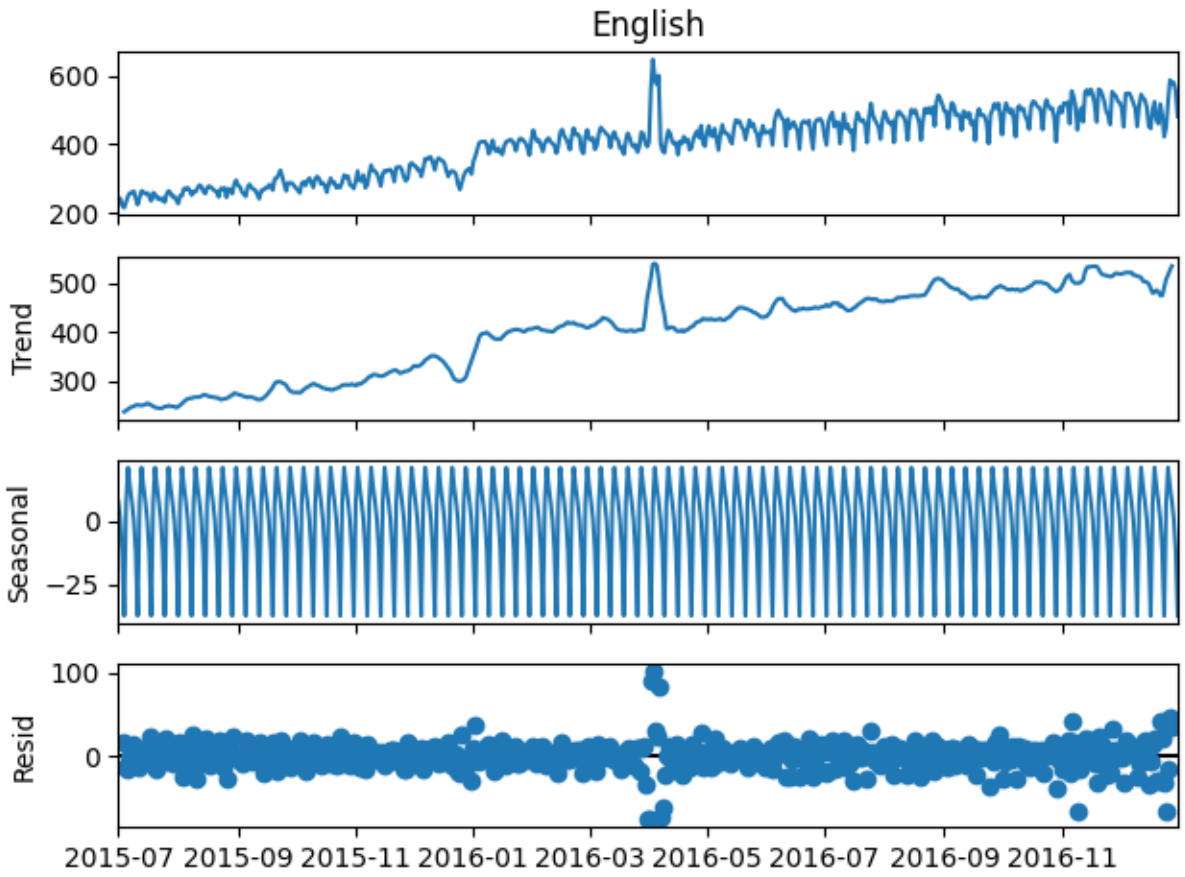


Figure 2.9: The decomposition sample of the English language

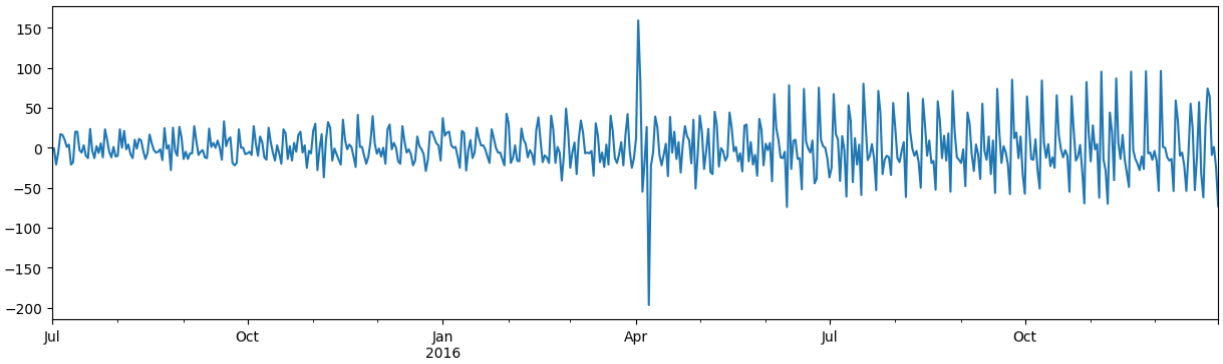


Figure 2.10: Differencing: time-series data "English"

With a 95 % confidence interval, the value of the English language is forecasted and illustrated in Figure 2.14

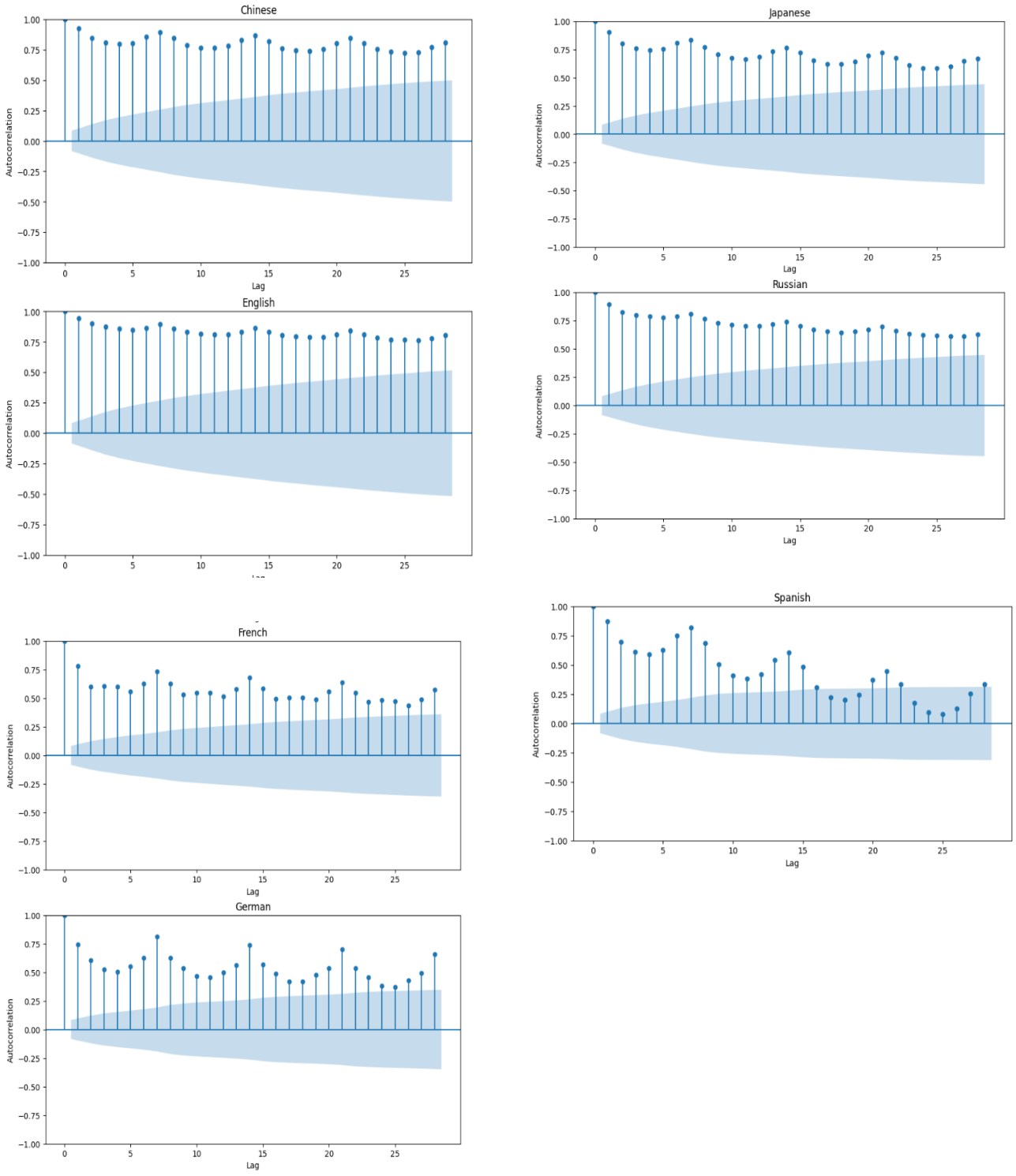
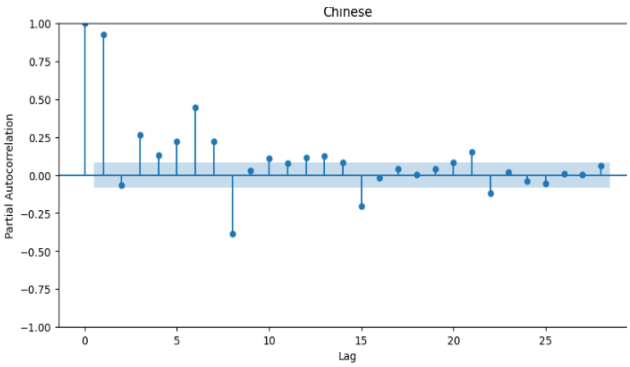
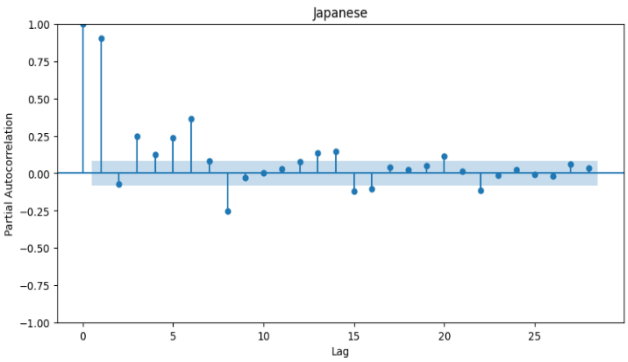
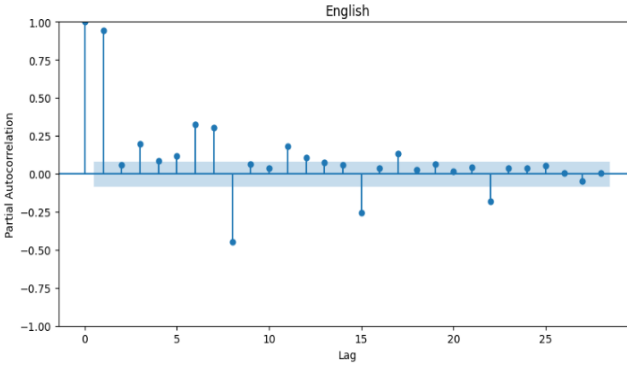
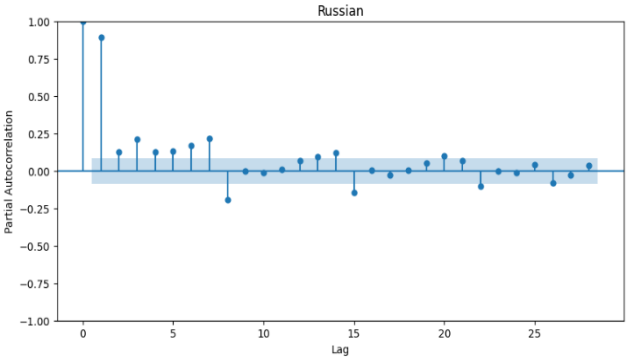


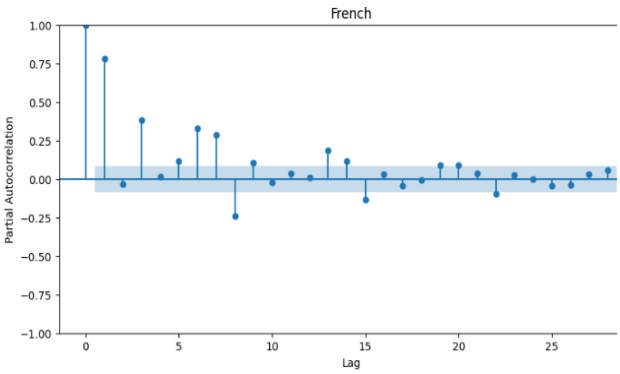
Figure 2.11: ACF plot of Chinese, English, French, German, Japanese, Russian, and Spanish language

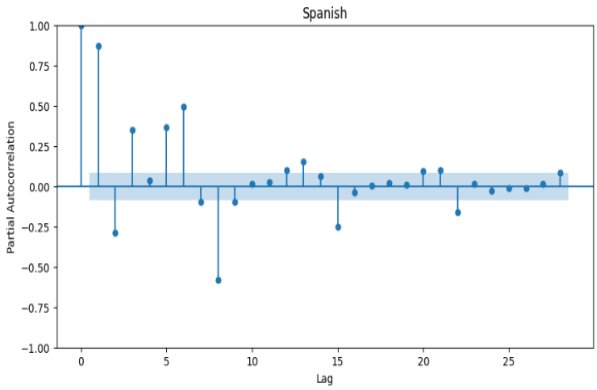

Chinese
Partial Autocorrelation
Lag


Japanese
Partial Autocorrelation
Lag


English
Partial Autocorrelation
Lag


Russian
Partial Autocorrelation
Lag


French
Partial Autocorrelation
Lag


Spanish
Partial Autocorrelation
Lag

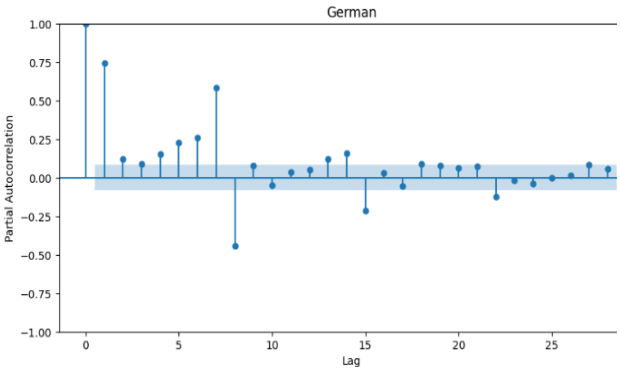

German
Partial Autocorrelation
Lag

Figure 2.12: PACF plot of Chinese, Japanese, English, Russian, French, Spanish and German language

Table 2.1 : MAE, RMS, and MAPE values for all languages using the ARIMA model

Language	Mean Absolute Error	Root Mean Square Error	Mean Absolute Percentage Error
1. Chinese	4.55	5.717	0.04
2. English	33.423	39.704	0.064
3. French	13.346	16.541	0.079
4. German	9.952	13.338	0.067
5. Japanese	11.004	13.824	0.045
6. Russian	27.686	33.206	0.081
7 Spanish	79.917	107.26	0.304

Table 2.2 : MAE, RMS, and MAPE values for all languages using the SARIMA model

Language	Mean Absolute Error	Root Mean Square Error	Mean Absolute Percentage Error
1. Chinese	4.668	5.834	0.043
2. English	30.146	39.148	0.061
3. French	12.546	15.907	0.076
4. German	9.269	11.916	0.064
5. Japanese	16.799	20.143	0.072
6. Russian	19.097	26.982	0.058
7 Spanish	80.541	110.725	0.309

4. Forecasting with the Facebook prophet

Forecasted the time series for the English language without using exog data and plotted in Figure 2.15. Forecasting for different languages done using Facebook Prophet without exog and the corresponding Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are summarized in Table 2.4.

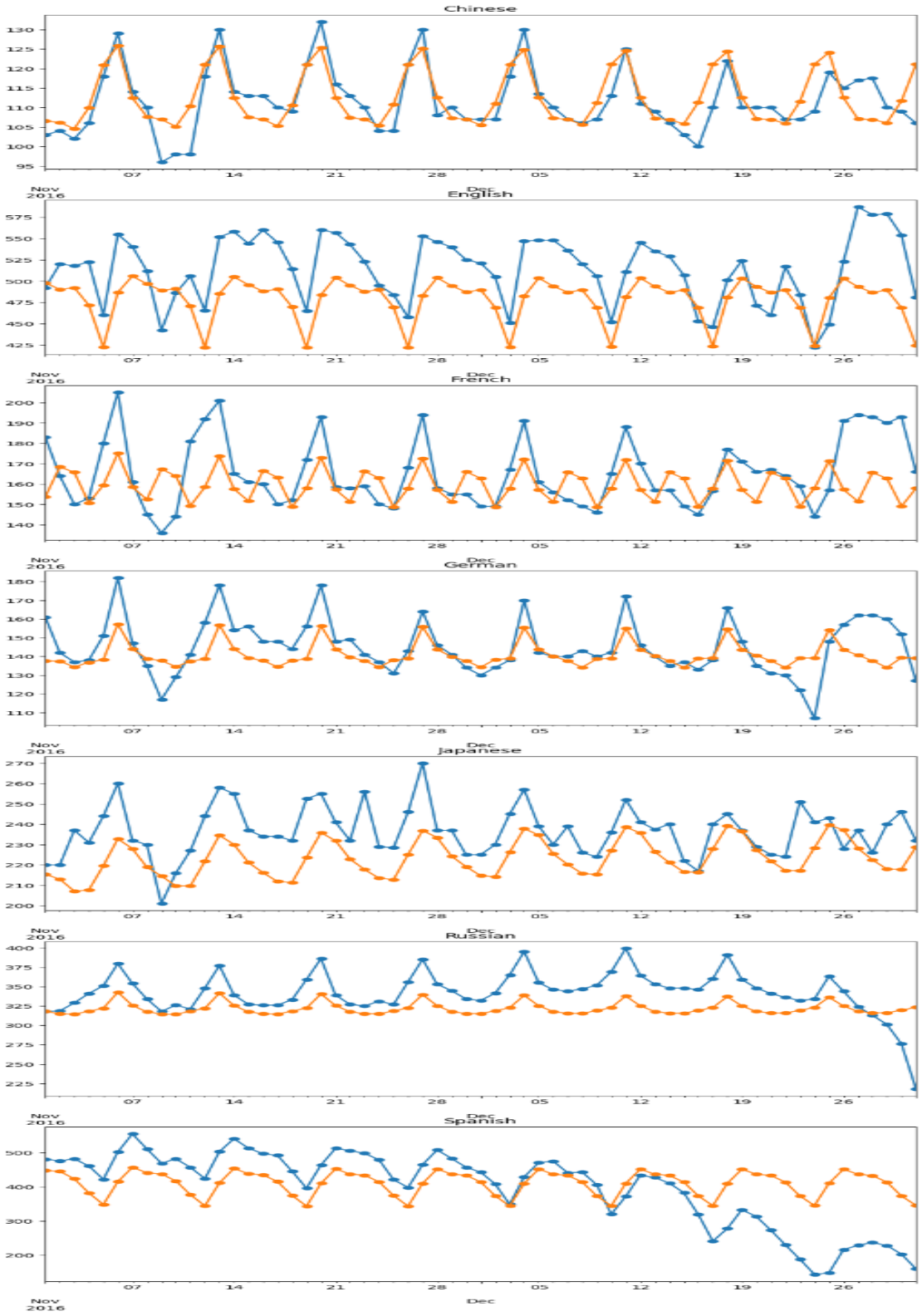


Figure 2.13: ARIMA model predicted and the actual time series for all languages

Table 2.3 : MAE, RMS, and MAPE for all languages using the SARIMAX model

Language	Mean Absolute Error	Root Mean Square Error	Mean Absolute Percentage Error
1. Chinese	4.514	5.526	0.041
2. English	25.959	34.745	0.053
3. French	13.282	17.804	0.078
4. German	9.406	12.155	0.065
5. Japanese	16.836	19.941	0.072
6. Russian	18.726	26.700	0.057
7 Spanish	75.725	94.918	0.269

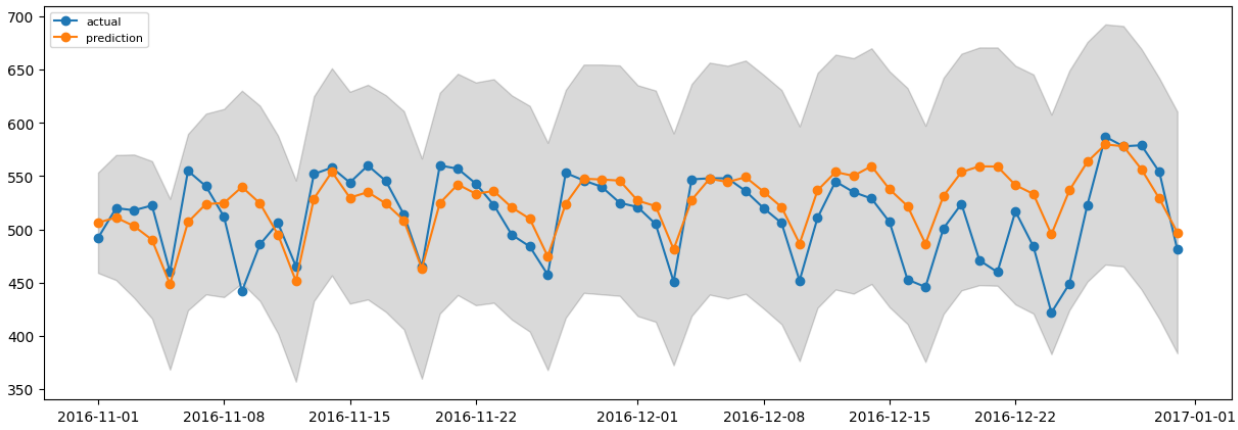


Figure 2.14: English language forecast (with a 95 % confidence interval)

Table 2.4 : MAE, RMS, MAPE using Facebook Prophet without exog data

Language	Mean Absolute Error	Root Mean Square Error	Mean Absolute Percentage Error
1. Chinese	4.332	5.475	0.04
2. English	25.565	31.885	0.05
3. French	14.35	16.283	0.089
4. German	8.825	11.565	0.061
5. Japanese	6.716	9.687	0.029
6. Russian	16.397	26.786	0.052
7 Spanish	75.066	111.321	0.302

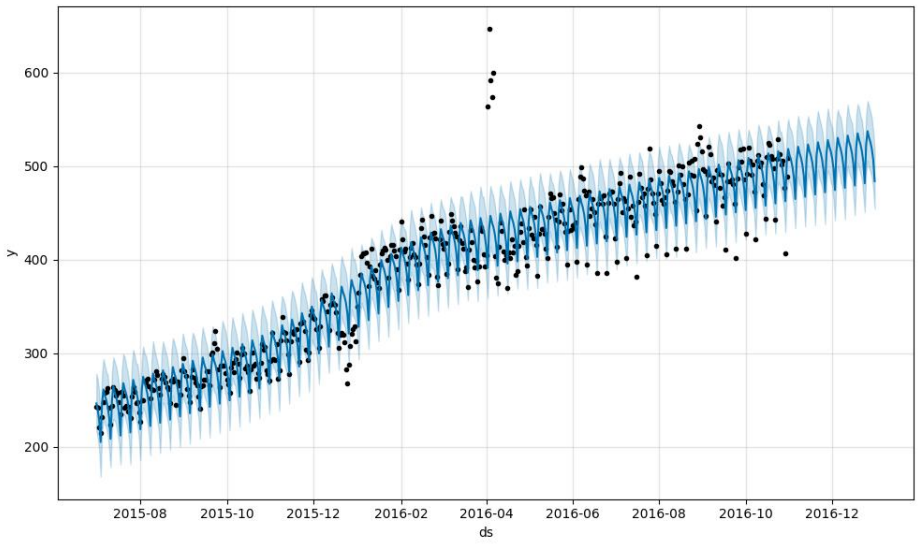


Figure 2.15: English language Forecast without using exog data

Forecasting for different languages done using Facebook Prophet with exog and the corresponding Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are summarized in Table 2.5.

Table 2.5 : MAE, RMS, and MAPE values for all languages using Facebook Prophet within exog data

Language	**Mean Absolute Error**	**Root Mean Square Error**	**Mean Absolute Percentage Error**
1. Chinese	333.93	336.273	3.014
2. English	69.699	75.903	0.135
3. French	280.176	283.4	1.723
4. German	300.34	302.849	2.095
5. Japanese	209.118	213.342	0.891
6. Russian	103.364	113.584	0.311
7 Spanish	75.066	111.321	0.302

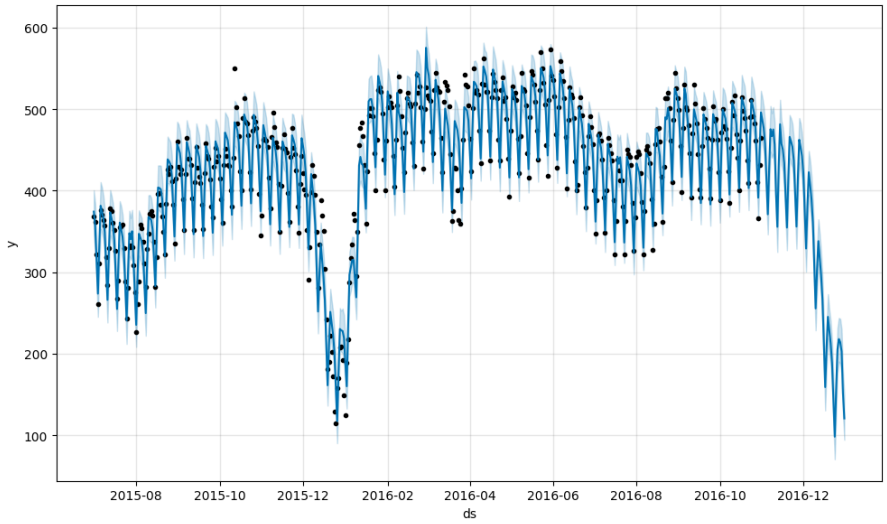


Figure 2.16: Forecasted the time series for the English language using exog data

Insights and Recommendations

Insights

1. **Access Types and Origins:**

   Access Types: The majority of the page views come from the "All Access" type, accounting for more than 50% of the total views. This indicates that users access Wikipedia through various platforms, including mobile and desktop.

   Access Origins: "All agents" access origin constitutes above 75% of the total views, suggesting that most traffic is generated by human users rather than automated bots.

2. **Language Distribution:**

   High Median Values: English and Spanish pages have the highest median values, indicating these languages attract more consistent traffic.

   Low Median Value: Chinese pages have the lowest median value, suggesting less frequent visits compared to other languages.

3. **Stationarity and Time Series Analysis:**

   Initial tests showed that all language time series were non-stationary. After differencing, the time series became stationary, which is crucial for accurate forecasting. Decomposition

of the time series helped in understanding the underlying trends, seasonal patterns, and residuals.

4. **Model Performance:**

ARIMA Model: Provided forecasts with varying accuracy across different languages. The performance metrics shown in Table 2.1 (MAE, RMSE, MAPE) indicate the model's effectiveness in capturing the patterns in the data.

SARIMA Model: Improved performance by incorporating seasonal components, leading to better forecasts for languages with strong seasonal patterns. The Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are summarized in Table 2.2.

SARIMAX Model: Further enhanced forecasts by including exogenous variables, which helped in capturing external influences on page views. The Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are summarized in Table 2.3.

Facebook Prophet: Showed robust performance both with and without exogenous variables, making it a versatile tool for forecasting. The Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) are summarized in Table 2.4.

Recommendations

1. **Resource Allocation:**

Allocate more server resources during peak times identified through forecasting to ensure website reliability and prevent downtime. Focus on optimizing infrastructure for "All Access" and "All Agents" types, as they constitute the majority of the traffic.

2. **Marketing Strategies:**

Utilize forecasted peak times to plan marketing campaigns and ad placements, maximizing visitor engagement and ad performance. Tailor marketing efforts based on language-

specific traffic patterns. For instance, prioritize English and Spanish pages for higher engagement.

3. **Ad Placement Optimization:**

   Use the forecasted data to predict ad performance on different language pages, helping clients optimize their ad placements. Consider the median values and traffic patterns of each language to strategically place ads where they are likely to get the most visibility.

4. **Model Selection:**

   For languages with strong seasonal patterns, prefer SARIMA or SARIMAX models to capture the seasonality effectively. Use Facebook Prophet for its flexibility and ability to incorporate external factors, providing robust forecasts even with complex data.

5. **Continuous Monitoring and Adjustment:**

   Regularly update the models with new data to maintain accuracy in forecasts. Monitor the performance metrics (MAE, RMSE, MAPE) to identify any deviations and adjust the models accordingly.

By implementing these insights and recommendations, the digital ad company can enhance their forecasting accuracy, optimize resource allocation, and improve ad performance, ultimately leading to better client satisfaction and business outcomes.

Business Questions Answered from Analysis

1. Defining the problem statements and where can this and modifications of this be used?

   The case study aims to understand the per-page view report for different Wikipedia pages for 550 days and forecast the number of views so that the digital ad company can predict and optimize the ad placement for their clients. The dataset consists of 145k Wikipedia pages and the daily view count for each of them. The clients belong to different regions and need data on how their ads will perform on pages in the various languages of the website.

There are two csv files given

1. train_1.csv: In the csv file, each row corresponds to a particular article and each column corresponds to a particular date. The values are the number of visits on that date.

The page name contains data in this format:

SPECIFIC NAME _ LANGUAGE.wikipedia.org _ACCESS TYPE_ACCESS ORIGIN

having information about the page name, the main domain, the device type used to access the page, and also the request origin (spider or browser agent)

- Exog_Campaign_eng: This file contains data for the dates that specify that particular date has a campaign or significant event that could affect the views for that day. The data is just for pages in English.

There's 1 for dates with campaigns and 0 for remaining dates. It is to be treated as an exogenous variable for models when training and forecasting data for pages in English

2. Write 3 inferences you made from the data visualizations

- English and Spanish have the higher median values
- Chinese has the lowest median value
- All Access type has the highest percentage of access type which is more than 50 %
- All agents access origin has the highest percentage of access origin type of above 75 %

3. What does the decomposition of series do?

- The decomposition of a time series involves breaking it down into its components. Namely: Trends, Seasonality
    - **Trend:** The long-term movement or direction in the time series. It represents the underlying pattern that shows whether the series is generally increasing, decreasing, or stable over time.
    - **Seasonality:** The repeating, periodic fluctuations in the time series that occur with a fixed frequency. For example, language English has a seasonal pattern

- **Cycle:** The longer-term undulating pattern in the time series that is not strictly tied to a fixed frequency. Cycles are often associated with economic or business cycles and can have varying durations.
- **Residuals (or Irregular Component):** The remaining variability in the time series that cannot be explained by the trend, seasonality, or cycle. It includes random fluctuations, noise, and any irregular patterns.

4. What level of differencing gave you a stationary series?

   A single level of differencing gives a stationary series

5. Difference between ARIMA, SARIMA & SARIMAX.

   ARIMA, SARIMA, and SARIMAX are all-time series forecasting models used in statistics and machine learning. Here's a breakdown of each model and their differences:

1. ARIMA (AutoRegressive Integrated Moving Average)

- **Components**:
  - **AR (AutoRegressive)**: The model uses the relationship between an observation and a number of lagged observations (previous time points).
  - **I (Integrated)**: This refers to the differencing of raw observations to allow for the time series to become stationary (i.e., mean and variance are constant over time).
  - **MA (Moving Average)**: The model uses the dependency between an observation and a residual error from a moving average model applied to lagged observations.
- **Use Case**: ARIMA is used for univariate time series data that is stationary or can be made stationary through differencing.

2. SARIMA (Seasonal ARIMA)

- Components:
  - SARIMA extends ARIMA by adding seasonal components to account for seasonality in the data.
  - It incorporates seasonal differencing along with seasonal AR and MA terms.
- Notation: SARIMA is denoted as ARIMA(p, d, q)(P, D, Q, s), where:
  - (P, D, Q): Seasonal components corresponding to the seasonal AR, differencing, and MA terms.

- s: The length of the seasonal cycle (e.g., s=12 for monthly data).
- Use Case: SARIMA is suitable for univariate time series data with seasonality.

3. SARIMAX (Seasonal ARIMA with eXogenous variables)

- Components:
  - SARIMAX builds on SARIMA by allowing for the inclusion of exogenous variables (external factors that might affect the time series).
- Notation: SARIMAX is denoted similarly to SARIMA, but includes additional parameters for exogenous variables, typically represented as X.
- Use Case: SARIMAX is useful for time series data that may be influenced by other factors, enabling the model to capture more complex relationships in the data.

Summary of Differences

- ARIMA: Best for non-seasonal, stationary time series data.
- SARIMA: Best for seasonal time series data without external variables.
- SARIMAX: Best for seasonal time series data with external variables included.

These models are widely used in time series analysis and forecasting, and the choice among them depends on the characteristics of the data being analyzed.

6. Compare the number of views in different languages The percentage of each language concerning total views is
   - English 16.62 %
   - Japanese 14.08 %
   - German 12.79 %
   - French 12.27 %
   - Chinese 11.88 %
   - Russian 10.36 %
   - Spanish 9.70 %
7. What other methods other than grid search would be suitable to get the model for all languages?
   - hyperparameter of the models can be observed from ACF and PACF plot, random search.