Project Report

Project Name:
Web Traffic Analysis (Santa Monica Government Websites)

Team Members:
Rahul Purushottam Gaonkar (rpg283)
Bhushan Manohar Newalkar (bmn258)

Problem:

We performed web traffic analysis across the City of Santa Monica's web presence. Web traffic analysis involved time series forecasting of web page views. The analysis will help the website owner to understand the traffic they are getting on the websites and will help them to take decisions regarding advertisements, website design, customer acquisition, server maintenance, etc.

Background:

We gathered information about web traffic analysis through various sources over the internet. The below list of publications was used to learn about the background of web traffic analysis:

https://www.researchgate.net/publication/272815693_Web_Analytics_Overview
https://www.researchgate.net/publication/281448685_Web_Analytics
https://pdfs.semanticscholar.org/1a33/231a1dfd922e463ad0e4c473af19f328543c.pdf
https://faculty.ist.psu.edu/jjansen/academic/jansen_website_analysis.pdf
https://www.instituteforpr.org/wp-content/uploads/Seth_Duncan_Web_Analytics.pdf

Data Description:

We have collected data from https://data.smgov.net/Public-Services/Web-Analytics/8dh4-6epx. This data is aggregated from Google analytics via analytics.smgov.net, as well as PDF requests parsed from server logs.

Attribute Name	Attribute Description	Attribute Data Type	Attribute Type	Attribute Semantic
date	The day on which these stats were captured	DateTime	Quantitative	Temporal

domain	The top-level website domain	string	Nominal	Hierarchical
page	The relative page URL for a page under the domain	string	Nominal	Hierarchical
page_title	The web page's title	string	Nominal	Hierarchical
percent_new_sessions	The percentage of sessions that were from new users	numeric	Quantitative	Sequential
pageviews	The number of visits to this page	numeric	Quantitative	Sequential
unique_pageviews	Multiple page views of a single page within a visitor's session count as a single unique page view	numeric	Quantitative	Sequential
avg_time_on_page	The average amount of time in seconds spent on this page	numeric	Quantitative	Sequential
avg_page_load_time	The average amount of time in seconds this page took to load	numeric	Quantitative	Sequential
entrances	The number of times visitors to the domain started on this page	numeric	Quantitative	Sequential
entrance_rate	The percentage of visitors to this domain that started on this page	numeric	Quantitative	Sequential
bounces	The number of times visitors viewed this page only and then left the domain	numeric	Quantitative	Sequential
bounce_rate	The percentage of visitors viewing this page only and then leaving the domain	numeric	Quantitative	Sequential
exits	The number of times visitors to this page left the domain	numeric	Quantitative	Sequential
exit_rate	The percentage of visitors to this page that left the domain	numeric	Quantitative	Sequential
bytes_sent	The total number of bytes sent for requests to this page (only available for PDF)	numeric	Quantitative	Sequential

id	unique row identifier, calculated from the date + domain + page	string	Ordinal	Sequential

1. The feature that will be used for forecasting is '**pageviews**'.
2. We are not using '**unique_pageviews**' feature for forecasting instead of '**pageviews**' because we want to consider the cases where a user visits the same page multiple times in the same session as different pageviews.
3. We are using features like **'bounces'**, **'avg_time_on_page'** and **'avg_page_load_time'** for further exploration. (Considering them as confounding variables and analyzing their effect on the '**pageviews**')

Project Approach:

Data Preprocessing:

1. The data contains the page views for several domains, which is a large amount of data and for this project a lot of it is unnecessary. Usually, a lot of internet users google for services and get directed straight away to the pages. Hence, we are considering only top 20 pages as per page views. This will help us to focus on the web pages that are usually getting many web page views.
2. For one of the pages in the top 20 pages namely **'bigbluebus.com/ /default.aspx'** we had a lot of pageviews value for different days as zero. So, we dropped that page and selected the 21st top page based on page views for our analysis.
3. We are only selecting features namely **'date', 'domain', 'page', 'page_title', 'pageviews', 'avg_time_on_page', 'avg_page_load_time'** and **'bounces'** for our analysis.
4. The inconsistencies in the domain names are removed. For e.g., some records have the domain name that starts with 'www.' while others do not. We also have a domain with the name 'localhost'.
5. We replaced all the NaN values as well as filled in missing date entries from '01-01-2017' to '11-10-2018' with the mean of the feature column considered.

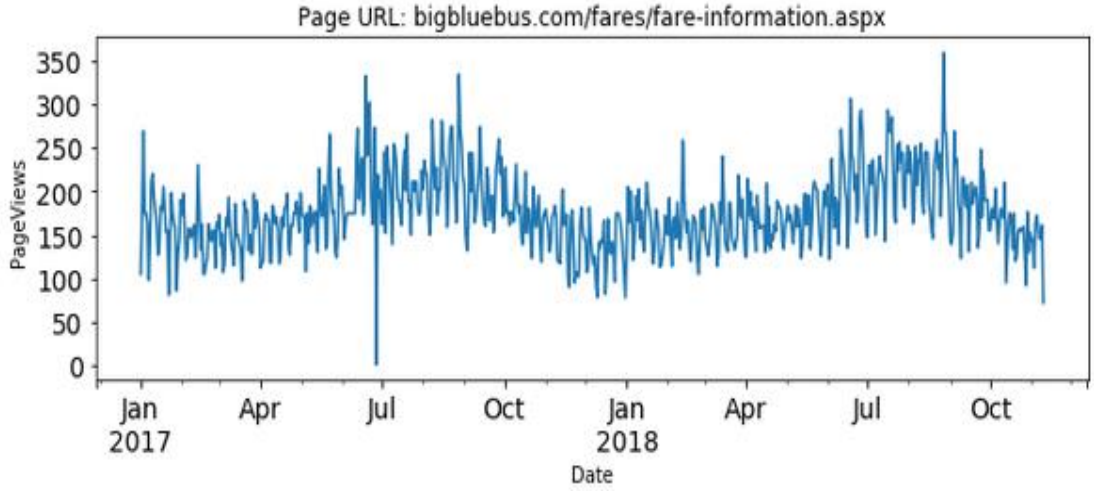
Steps in Model building:

1. **Testing Stationarity of Time Series Data:**
    i. We plotted the time series data to visually asses the trend and seasonality in the data.
    ii. We also plotted the summary statistics like rolling mean and rolling standard deviation of the time series data to get a better idea about the presence of trend and seasonality.
    iii. We also used Durbin-Watson statistic to test the stationarity of the time series

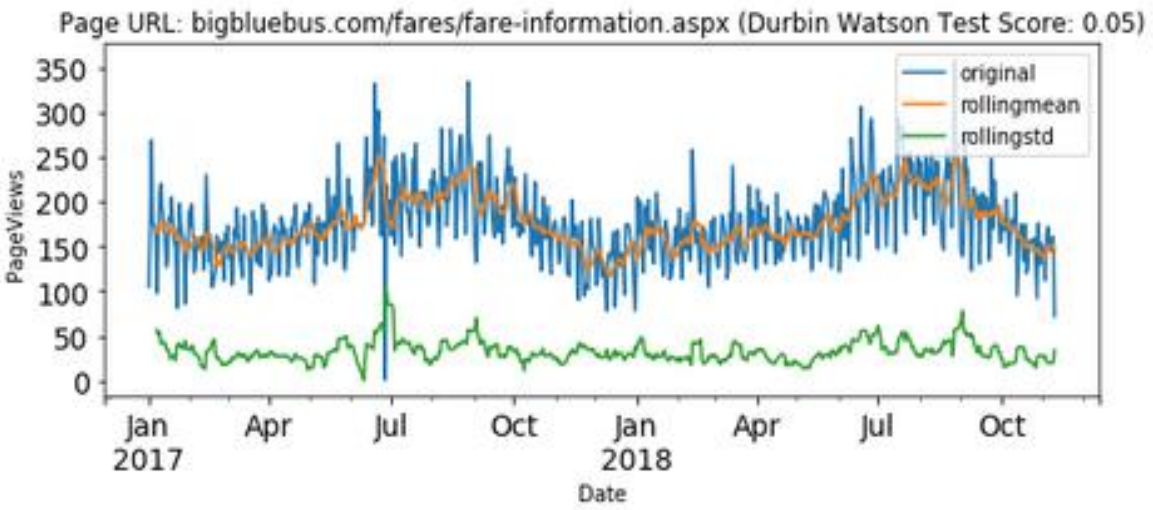
data. The Durbin-Watson statistic has a value between 0-4 where a value of 2 indicates no auto-correlation, value between 0-2 indicates positive auto-correlation and value between 2-4 indicates negative auto-correlation.

iv. We used the '**statsmodels.tsa.seasonal.seasonal_decompose**' package to find the seasonal component in the time series data and found out that the seasonal component is of 7 days i.e. week. (Possible reason for this could be as these are government websites which are mainly used for formalities like paying bills, taxes or access some government services which people tend to do more on weekends)

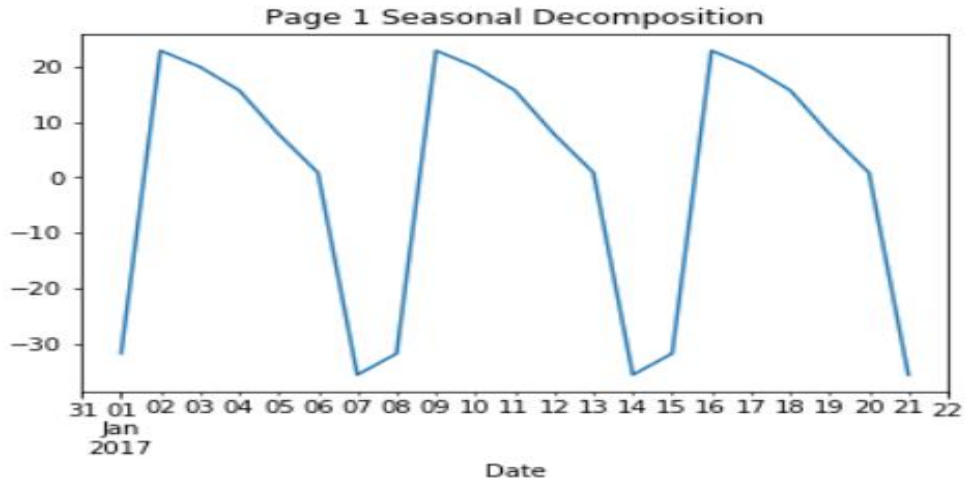
- The time series data plot:



- Rolling mean and Rolling standard deviation of the time series data plot:

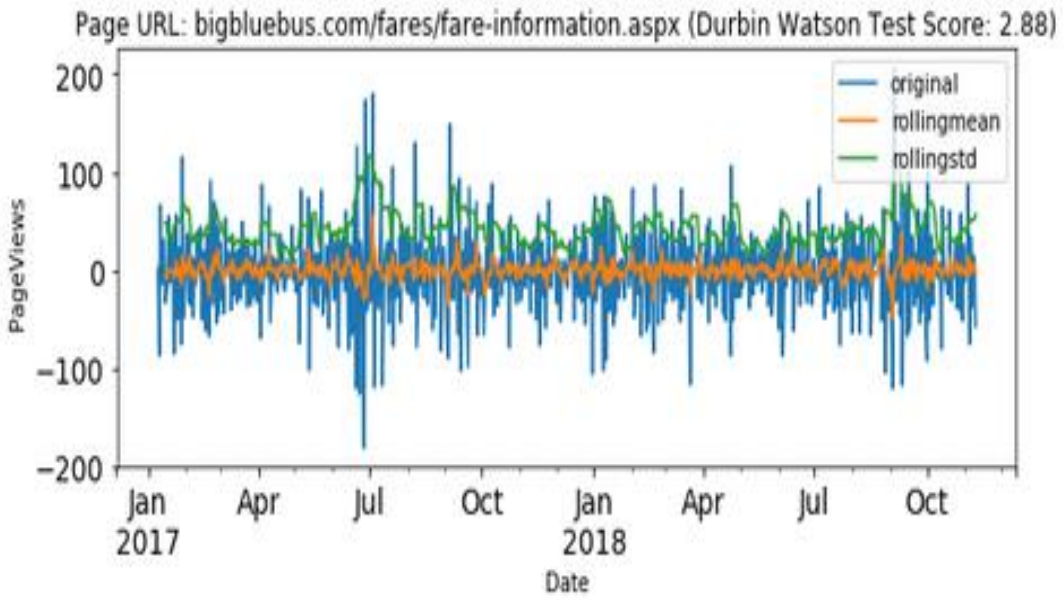


- Seasonal decomposition:



v. To remove trend and seasonality from the time series data we performed first difference of seasonal difference of the time series data. This stabilized the rolling mean and rolling standard deviation of the time series data thereby improving the stationarity of the time series data.

- Time series data, rolling mean and rolling std deviation after differencing:



2. Identification of Model:

i. We selected **SARIMAX** model (Seasonal ARIMA) as our data was non-stationary and had a seasonal component. We observed that ARIMA (or ARMA) model doesn't perform well for seasonal data (or non-stationary data).

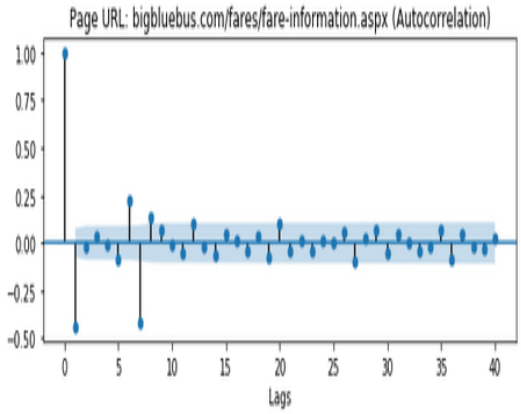
ii. We plotted ACF and PACF graph for all the 20 web pages and identified the AR, MA, SAR, SMA terms of SARIMAX model for every page using the Box and Jenkins method.

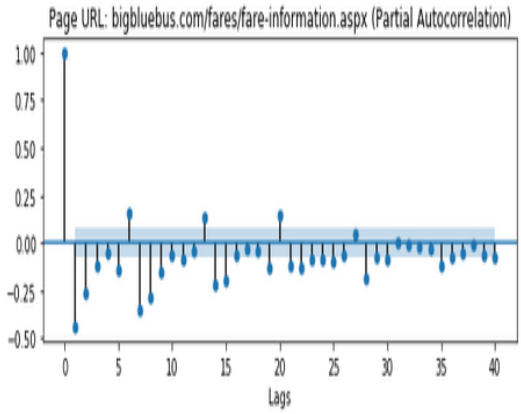
For e.g.

Page URL: bigbluebus.com/fares/fare-information.aspx

AR Term: 0 MA Term: 1 SAR Term: 1 SMA Term: 1

- ACF and PACF plots:





3. Training and Evaluating the Model:

i. We used **TimeSeriesSplit Cross Validator** to train and validate the SARIMAX model in order to prevent the forward bias as well as promote generalization while training the SARIMAX model.

ii. We plotted the residuals generated after fitting the SARIMAX model, tested the null hypothesis and constructed a qq-plot to verify that the residuals come from a normal distribution i.e. white noise.

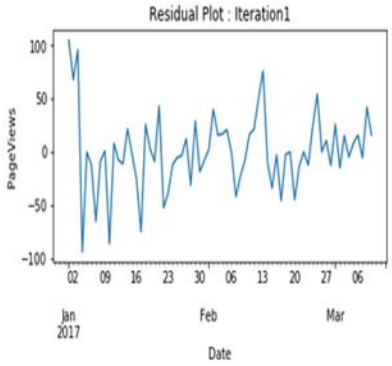
iii. We used forecast method of SARIMAX model which provided **Out-of-sample forecasts** i.e. predicting pageviews of samples outside of the training dataset. Thereby, preventing forward bias.

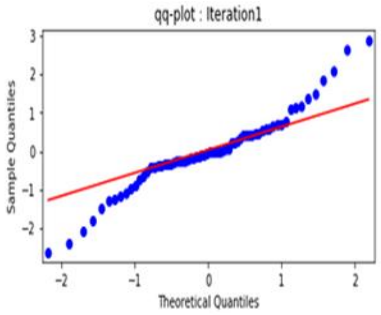
iv. We evaluated our SARIMAX model using the forecast error metric i.e. MAE and MFE.

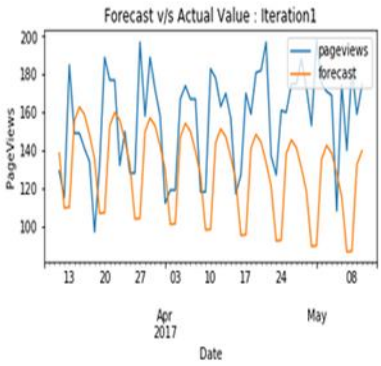
Evaluation Metric:

1. Mean absolute error: The mean absolute error (MAE) value is computed as the average absolute error value. If MAE is zero, the forecast is perfect.
2. Mean Forecast error: The MFE is the average error in the observations. A large positive MFE means that the forecast is undershooting the actual observations. A large negative MFE means the forecast is overshooting the actual observations.

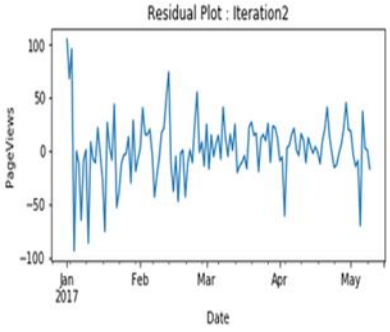
- Residuals plot, qq-plot and Forecast vs Actual Value plot for each cross-validation iteration of a page:

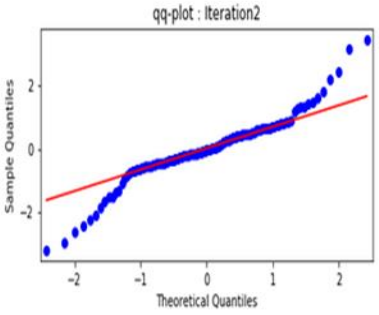


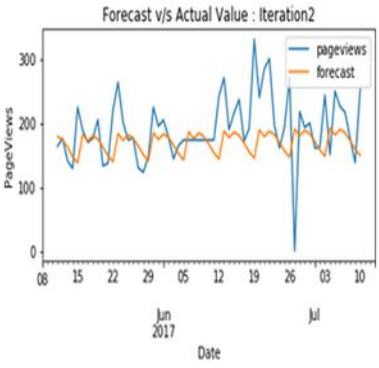




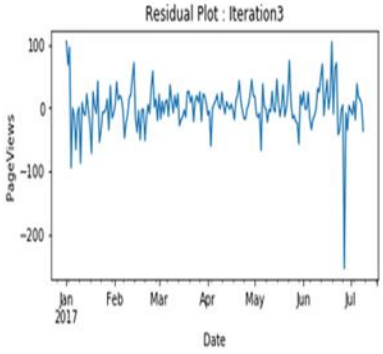
NormaltestResult(statistic=4.250023124063064, pvalue=0.11943158738695744)
Mean Absolute Error: 33.40
Mean Forecast Error: 27.43

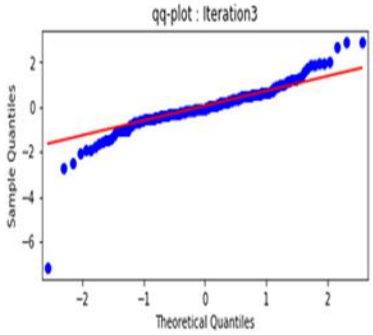


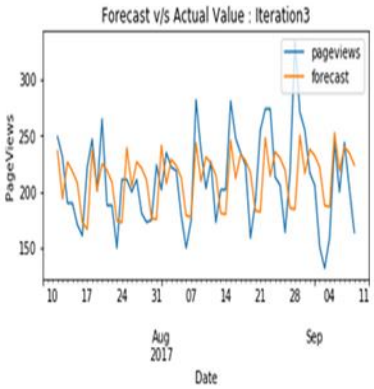




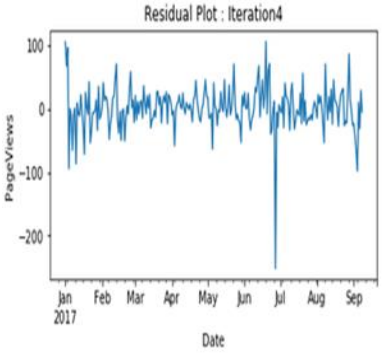
NormaltestResult(statistic=11.820911141988553, pvalue=0.002710951575117278)
Mean Absolute Error: 38.18
Mean Forecast Error: 21.88

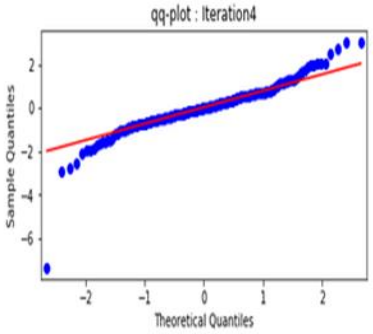


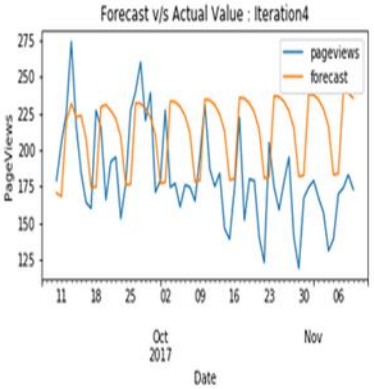




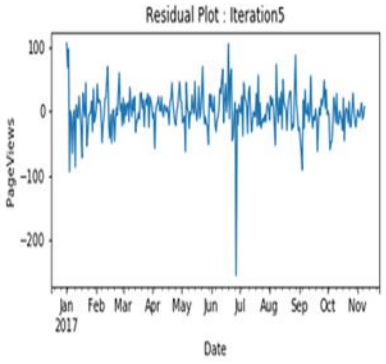
NormaltestResult(statistic=106.9333986758914, pvalue=6.021539116953224e-24)
Mean Absolute Error: 30.38
Mean Forecast Error: -2.13

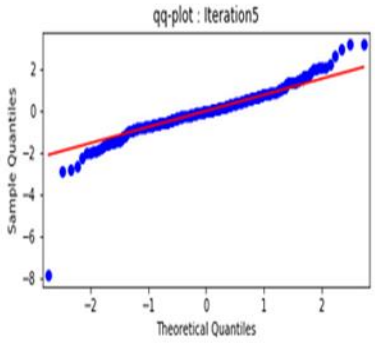


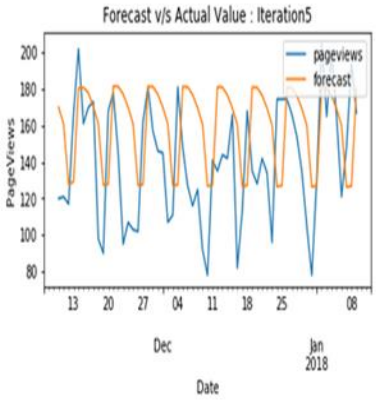




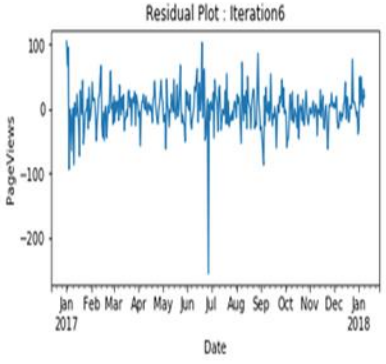
NormaltestResult(statistic=114.78457293071895, pvalue=1.1880827334795286e-25)
Mean Absolute Error: 41.36
Mean Forecast Error: -30.15

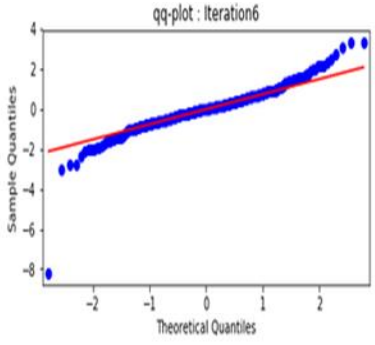


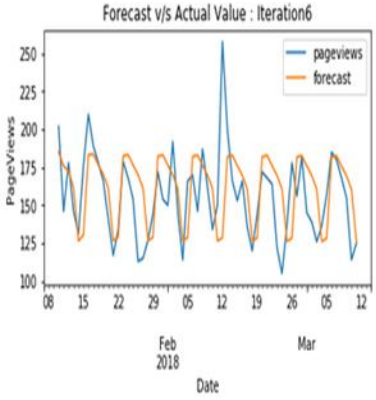




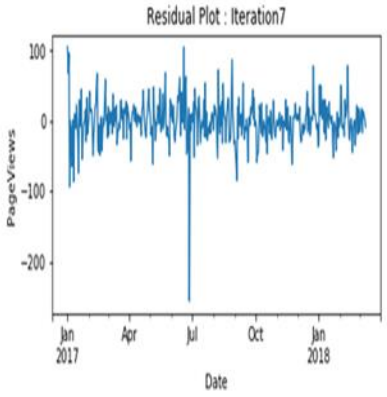
NormaltestResult(statistic=135.44278207549868, pvalue=3.8812674088905 86e-30)
Mean Absolute Error: 35.03
Mean Forecast Error: -19.33

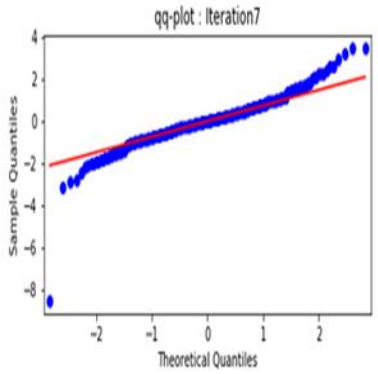


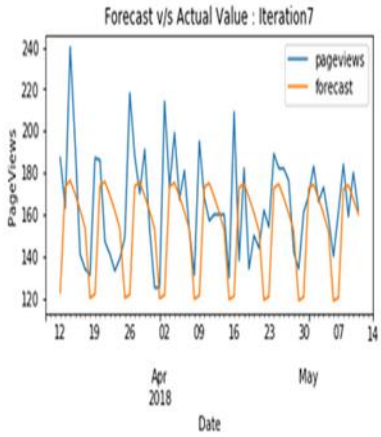




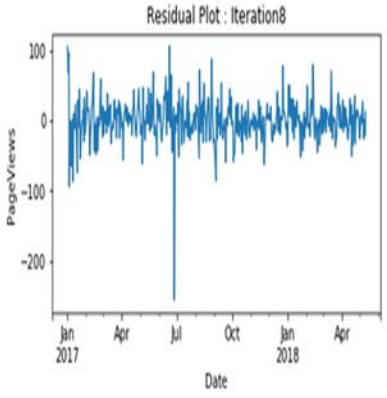
NormaltestResult(statistic=153.75515804279684, pvalue=4.097242311260848e-34)
Mean Absolute Error: 21.29
Mean Forecast Error: -5.36

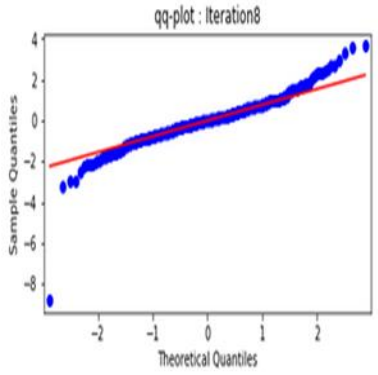


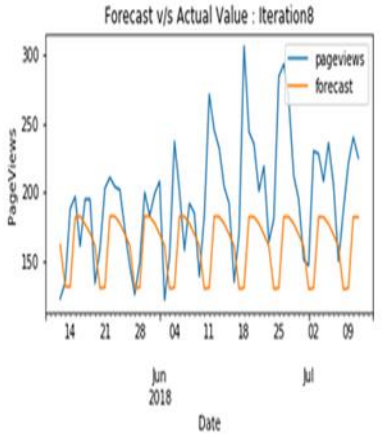




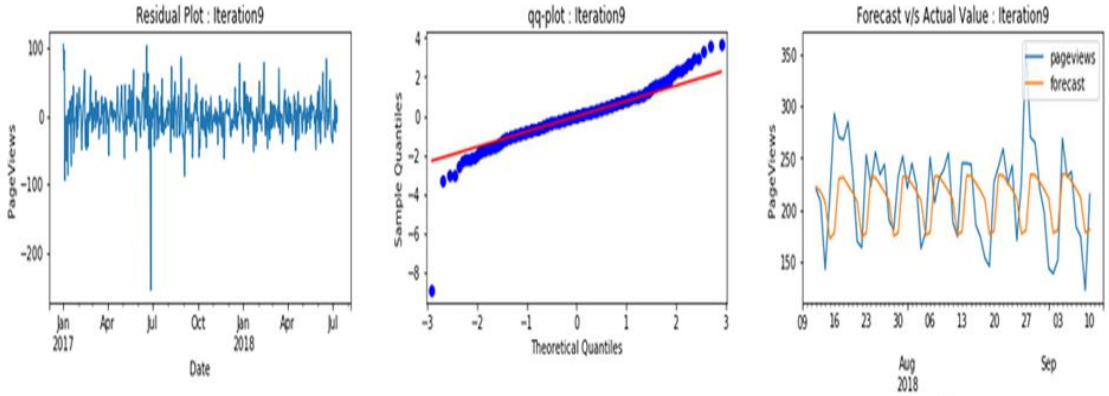
NormaltestResult(statistic=169.23822019881143, pvalue=1.7798664641937417e-37)
Mean Absolute Error: 22.78
Mean Forecast Error: 11.43



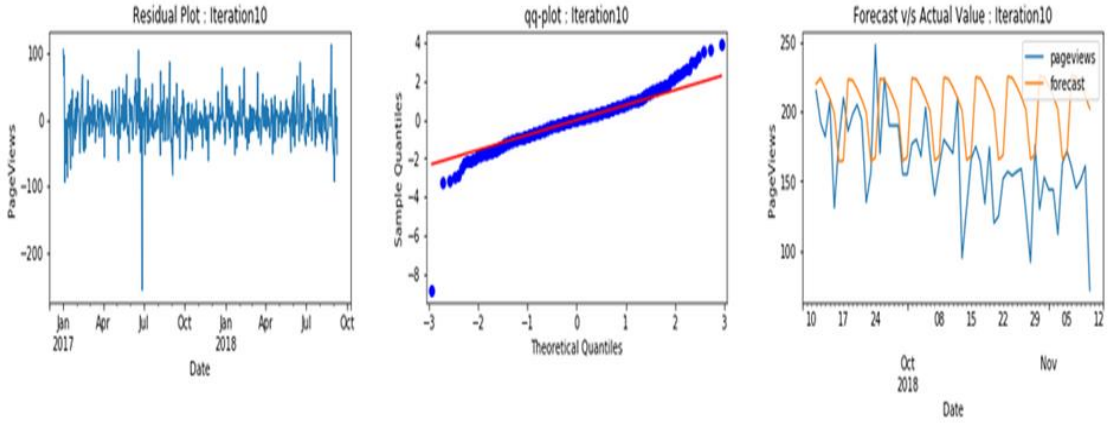




NormaltestResult(statistic=183.83781231108532, pvalue=1.2026140519770 57e-40)
Mean Absolute Error: 41.96
Mean Forecast Error: 34.10



NormaltestResult(statistic=191.28229812227858, pvalue=2.90788160301395e-42)
Mean Absolute Error: 32.99
Mean Forecast Error: 7.77



NormaltestResult(statistic=180.99432236754, pvalue=4.984048616698353e-40)
Mean Absolute Error: 43.24
Mean Forecast Error: -38.27

v. We can see that the residuals generated from different cross validation iterations of the top 20 web pages come from normal distribution and are evenly distributed which can be viewed through a qq-plot.

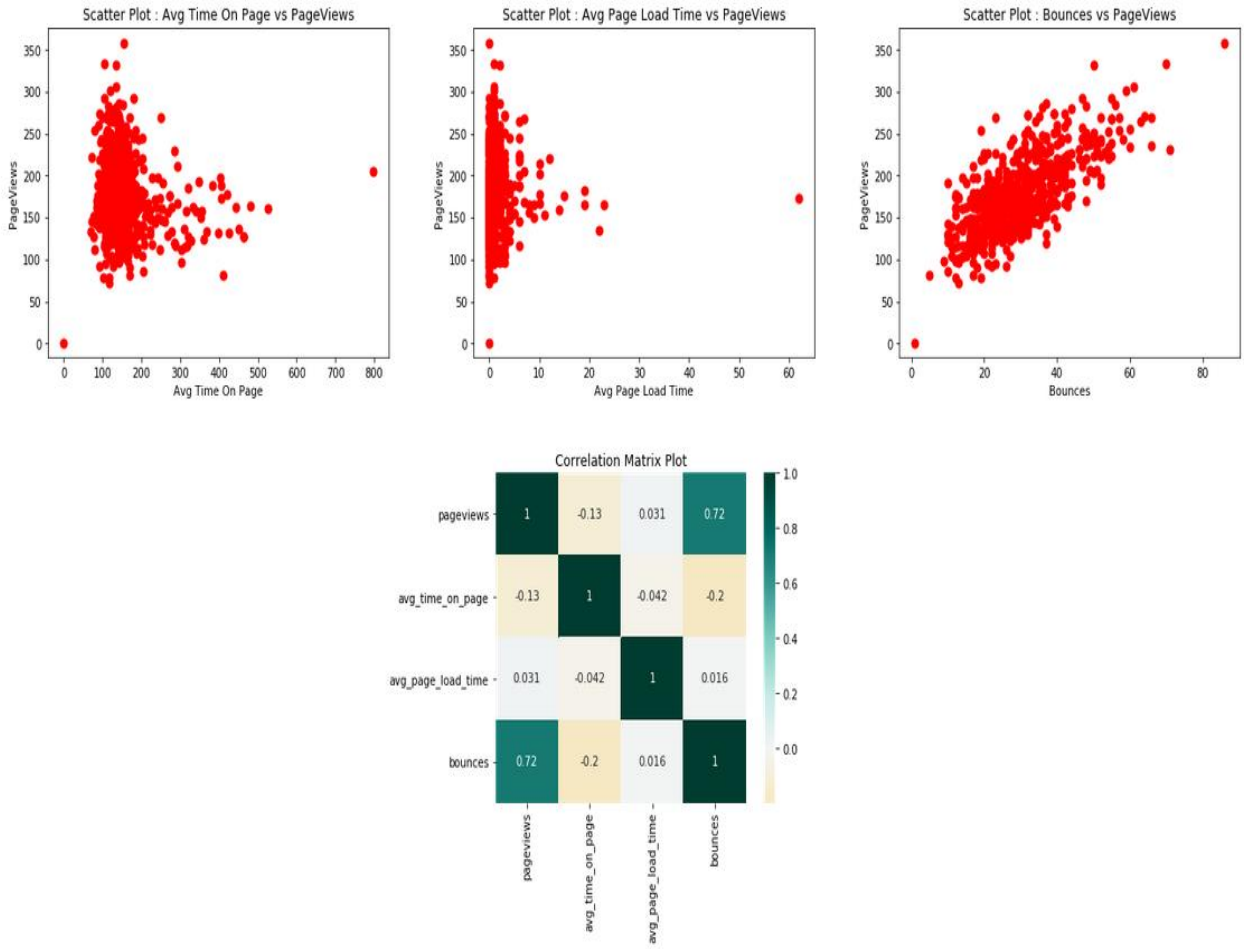
Effect of Confounding Variables on PageViews:

We analyzed the effect of the confounding variables on the page views of the web pages. General Assumptions regarding the confounding variables that can create a bias in time series modeling are as follows:

1. **avg_page_load_time:** The high value of average page load time can discourage the users to view the page in future i.e. decreasing the number of future page views.
2. **avg_time_on_page:** The high average time spent on the page indicates that the web page content is relevant to user interest and will encourage the user to view the page in future i.e. increasing the number of future page views. It can also help us to determine if user has been redirected to this page (by clicking on some advertisement) if the value of avg_time_on_page variable is low i.e. indicative of decreasing number of future page views.
3. **bounces:** The bounces variable can help you identify if a user or bot is viewing the web page and this will affect the number of page views. It can also help us to determine if user has been redirected to this page (by clicking on some advertisement).

We visualized the correlation of the confounding variables with the target variable i.e. the pageviews using the scatter plot. We also visualized the Pearson's correlation coefficient of the confounding variables with the target variable i.e. the pageviews and amongst themselves using heat map.

- Scatter Plot and Heat Map:



Relationship of confounding variables with PageViews:

1. **avg_time_on_page:** We can see that there is a positive as well as negative correlation between Avg Time On Page and PageViews, indicating that when the Avg Time On Page increases the PageViews increases and decreases respectively.
   a) Possible reason for positive correlation: The web page has a lot of content relevant to the user which engages the user viewing the web page. This can also motivate the user to view the page again in future thereby increasing the PageViews.
   b) Possible reason for negative correlation: The web page doesn't have content relevant to the user which results in the bouncing of the user viewing the web page to some other page.

2. **avg_page_load_time:** We can see that there is a positive as well as negative correlation between Avg Page Load Time and PageViews, indicating that when the Avg Page Load Time increases the PageViews increases and decreases respectively.
    a) Possible reason for positive correlation: The Web traffic analysis that we are performing is for government websites of Santa Monica which can have some useful and important information for the users. Therefore, even if the Avg Page Load Time of the web page is high the user will view that page thereby increasing the PageViews of that page.
    b) Possible reason for negative correlation: If the Avg Page Load Time value of the web page increases it will discourage the user from viewing that web page.
3. **bounces:** We can see that there is a positive correlation between Bounces and PageViews indicating that when the PageViews increase the Bounces also increase.
    Possible reasons for this are as follows:
    a) The user is redirected to this page from any top search result of search engines like Google.
    b) The user is redirected to this page from any other user relevant web page of the same website.
    c) The user is redirected to this page by clicking an advertisement on any other page.
    d) The bot is trying to access the web page.

Changelog:

The following things were added/updated in the Project Proposal to get the final draft of Project Report based on the feedback received:

1. We changed the Problem Statement as it was a bit vague.
2. We described the reason for selecting the target variable i.e. pageviews for forecasting. (We are not using 'unique_pageviews' feature for forecasting instead of 'pageviews' because we want to consider the cases where a user visits the same page multiple times in the same session as different pageviews.)
3. We finally selected the SARIMAX model for our project as it was good for non-stationary data having seasonal component.
4. We expanded every module written in Project Proposal with all the details and findings we got after completing the project.