Web Traffic Forecasting

Time Series Analysis: Group Project

Instructor:Prof. Amit Mitra

Members:

1)Abhishek Choudhary 210037
2)Jayant Soni 210468
3)Tanmey Agarwal 211098
4)Yash Verma 211197

About The Project

The training dataset consists of approximately 550 entries. Each of these entries represent the sum of daily views of a set of different Wikipedia articles.The original dataset contained daily information about views for 550 days of about 1,45,000 different wikipedia articles which was further simplified into a dataset with 550 entries as mentioned above.

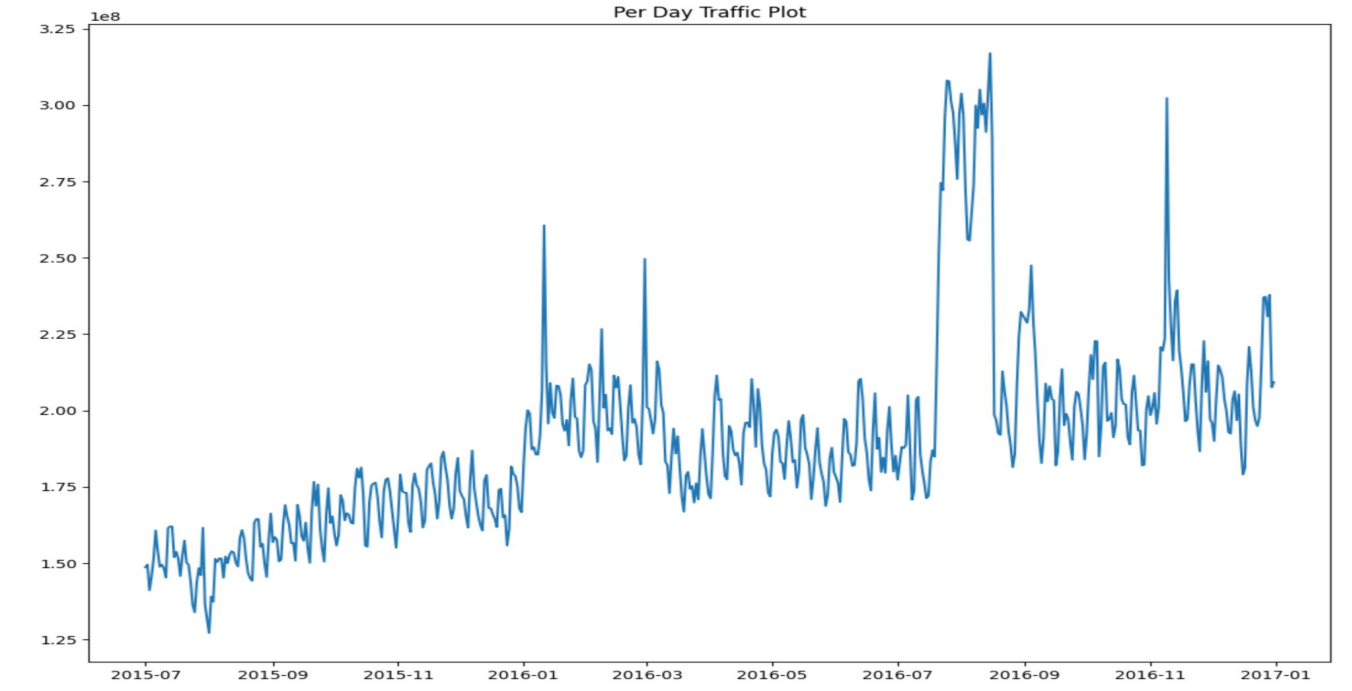
The dataset was divided into 2 parts - Training set containing 500 entries and the testing set containing 50 entries, and the goal was to predict 50 entries after training the model on the training set and measure it against the 50 values of the testing set.

We first checked the trend and seasonality present in this time series using Relative Ordering Test and Seasonal Decomposition Method respectively.

Relative Ordering Test led to the conclusion that a significant trend was present in the given time series data.

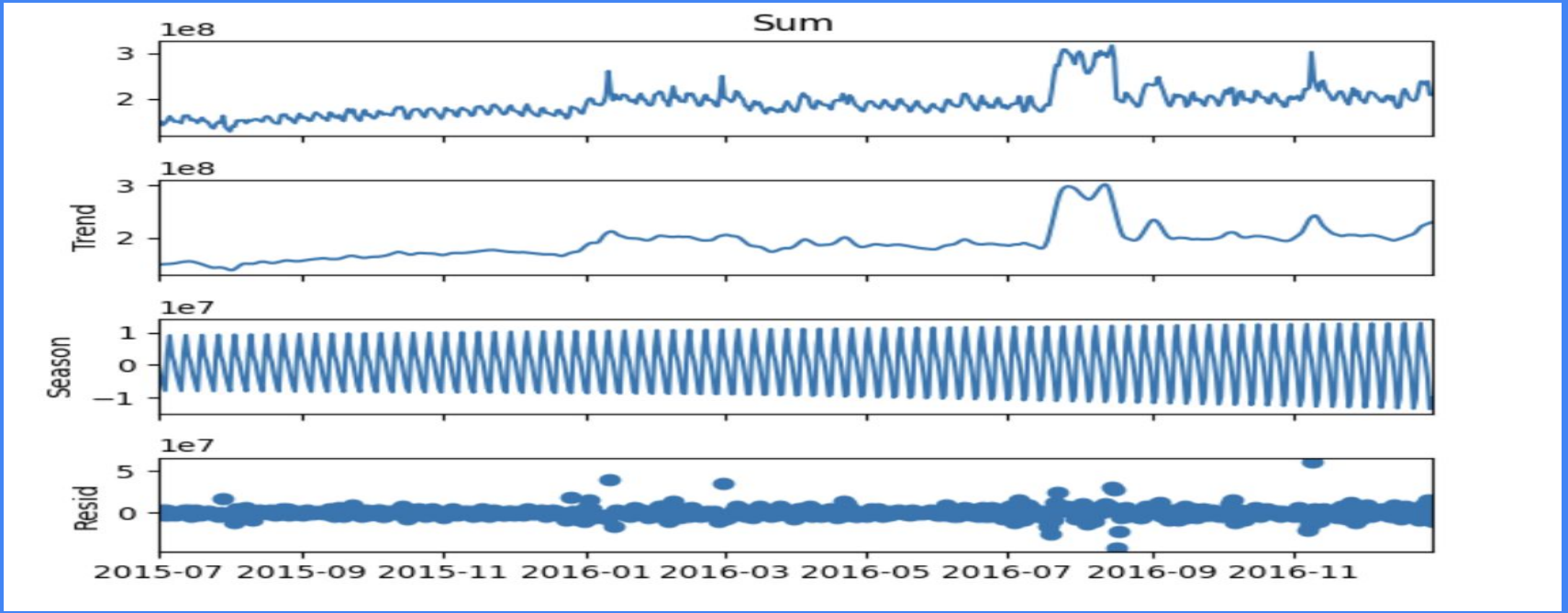
And seasonality was obvious after seeing the plot of the data set against the time frames.

Plot of the dataset (Per Day Traffic Plot) :



About The Project

In this graph Trend and Seasonality is shown individually.



Intro

After checking for trend and seasonality,we found them significantly.Moving forward we checked for the stationarity in the data using ADF(**Augmented Dickey Fuller)** test.
The null hypothesis of the ADF test is that the time series has a unit root, which indicates non-stationarity. The alternative hypothesis is that the time series is stationary. The test produces a test statistic and a p-value, and the decision to reject or fail to reject the null hypothesis is based on comparing the p-value to a chosen significance level (commonly 0.05).

If p-value<=0.05 then we conclude that the series is stationary.

```python
from statsmodels.tsa.stattools import adfuller
result = adfuller(train_data["Sum"])
print('ADF Statistic:', result[0])
print('p-value:', result[1])
print('Critical Values:', result[4])
#given data is not stationary
```

```
ADF Statistic: -2.228348845251733
p-value: 0.19611779063112333
Critical Values: {'1%': -3.443905150512834, '5%': -2.867517732199813, '10%': -2.569953900520778}
```

```python
result = adfuller(train_data["Sum"])
print('ADF Statistic:', result[0])
print('p-value:', result[1])
print('Critical Values:', result[4])
#first order differencing gave us stationary time series
```

```
ADF Statistic: -7.1604425030521615
p-value: 2.977751414147174e-10
Critical Values: {'1%': -3.443905150512834, '5%': -2.867517732199813, '10%': -2.569953900520778}
```

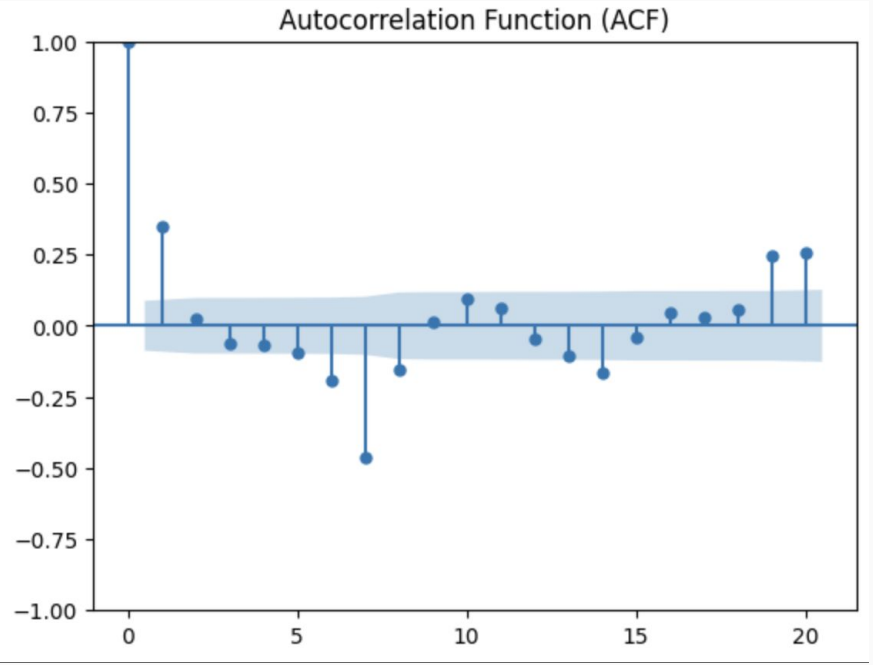
Checking stationarity :

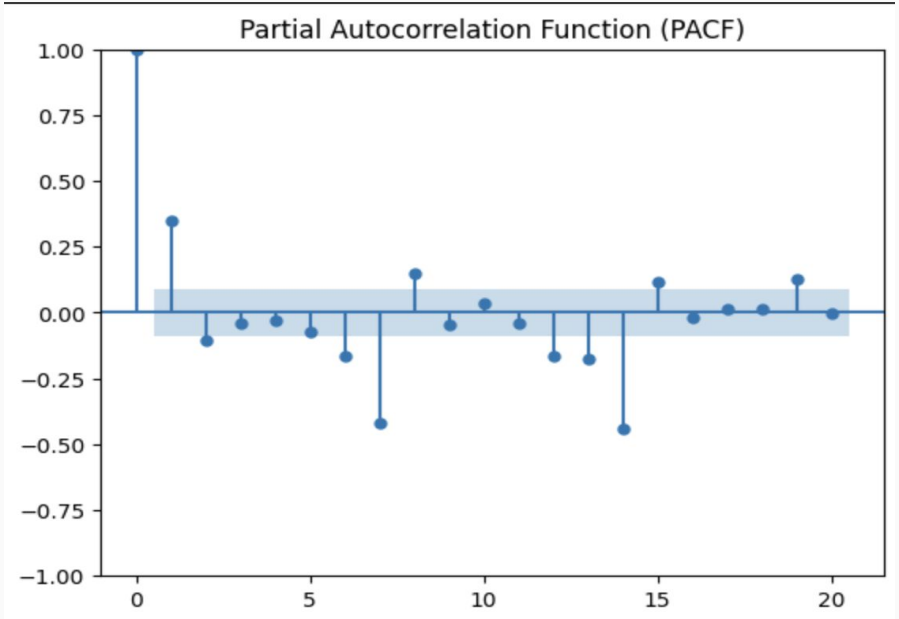
Initially the original dataset was not stationary. Applying differencing once gave us a time series which was covariance stationary since the p value was less than 0.05 (taking confidence level 95%).

We removed trend and seasonality from the series.

The ACF and PACF plots of the residual series removing time trend and seasonal trend are:

ACF And PACF Plots of Residual Time Series





Inferences from ACF and PACF

1) In the PACF plot, we observe that it decays gradually and there is a spike and a drop off after lags 1 and 2, hence we can make a good assumptions that p value (if AR part is present) is either 1 and 2.
2) In the ACF plot, we observe again that it decays gradually and cuts off after the first lag(q value can be 1).

Both ACF and PACF decay gradually hence we considered an ARMA model for the time series with p=1,2 and q=1 (A good and reasonable guess)

Implementing an ARMA/ARIMA model

After considering the inferences made from the ACF and PACF plots, we tried to fit an ARIMA model for the given time series data. We used the auto_arima function with the residual time series as the dataset and information criterion taken as BIC (Bayesian Information Criterion).

BIC = -2*loglikelihood + d*log(N), d=total number of parameters, N=sample size of training set. Generally a lower BIC model indicates a better model.

ARIMA parameters (p,d,q) - :

The model parameters after the previous step, specifically the order came out to be (2,0,1).

p=2 was concurrent with the conclusion we derived from the PACF plot of residual time series

q=1 was not obvious from the ACF plot but BIC worked as a confirmation.

d=0 is obvious since the residual time series does not require any differencing.

Fitting the model and prediction

```
model_bic = auto_arima(residual, suppress_warnings=True, seasonal=True, information_criterion='bic')
model_bic.order
```

```
(2, 0, 1)
```

```
model=model_bic.fit(residual)
predictions=model.predict(n_periods = 50)
predictions
```

```
final_predictions = np.array(predictions)+np.array(result_full.trend)[500:]+np.array(result_full.seasonal)[500:]
```

Model Parameters:

```
model.summary()
```

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	500
Model:	SARIMAX(2, 0, 1)	**Log Likelihood**	-8479.010
Date:	Mon, 13 Nov 2023	**AIC**	16968.019
Time:	12:53:20	**BIC**	16989.092
Sample:	0 - 500	**HQIC**	16976.288
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	-3.937e+04	2.46e+04	-1.600	0.110	-8.76e+04	8844.315
ar.L1	1.2422	0.035	35.384	0.000	1.173	1.311
ar.L2	-0.4397	0.037	-12.007	0.000	-0.511	-0.368
ma.L1	-0.9445	0.029	-32.568	0.000	-1.001	-0.888
sigma2	3.426e+13	0.000	2.12e+17	0.000	3.43e+13	3.43e+13

Ljung-Box (L1) (Q):	0.16	**Jarque-Bera (JB):**	933.04
Prob(Q):	0.69	**Prob(JB):**	0.00
Heteroskedasticity (H):	7.76	**Skew:**	0.33
Prob(H) (two-sided):	0.00	**Kurtosis:**	9.66

Final Predictions