Table of Contents

[THE CONCEPT AND APPLICATION OF TIME SERIES ANALYSIS USING MACHINE LEARNING MODELLING. 2](#_Toc151887087)

[DATA SOURCE 2](#_Toc151887088)

[APPLICATION OF AN APPROPRIATE BOX-JENKINS MODEL TO THE CHOSEN DATASET 2](#_Toc151887089)

[EXPLORATORY DATA ANALYSIS (EDA) 3](#_Toc151887090)

[THE AUGMENTED DICKEY-FULLER TEST 5](#_Toc151887091)

[RESULT INTERPRETATION 5](#_Toc151887092)

[TIME SERIES MODELLING. 6](#_Toc151887093)

[AUTOCORRELATION 7](#_Toc151887094)

[PARTIAL AUTOCORRELATION 8](#_Toc151887095)

[ARIMA MODEL 9](#_Toc151887096)

[**FORECAST** 10](#_Toc151887097)

[SARIMAX 11](#_Toc151887098)

[TEXT ANALYTICS 13](#_Toc151887099)

[MODELLING 14](#_Toc151887100)

[ELEMETS OF DASHBOARD 17](#_Toc151887101)

# 

# THE CONCEPT AND APPLICATION OF TIME SERIES ANALYSIS USING MACHINE LEARNING MODELLING.

Time series analysis is the endeavor of extracting meaningful summary and statistical information from points arranged in chronological order. It is done to diagnose past behavior as well as to predict future behavior. Time series and its data are increasingly important due to the massive production of such data through, for example, the internet, the digitalization of healthcare, and the rise of smart cities. (Aileen Nielsen,2020)

Time series analysis often comes down to the question of causality: how did the past influence the future? A variety of discipline have contributed novel ways of thinking about time series data set and applications. Time series data set have been driven by medicine, Forecasting weather, and forecasting economic growth. (Aileen Nielsen,2020)

Time series analysis using machine learning models involves selecting appropriate algorithms based on the characteristics of the data, handling temporal dependencies, and effectively capturing patterns to make accurate predictions or uncover meaningful insights. The choice of model depends on the specific characteristics and goals of the time series dataset at hand.

## **DATA SOURCE**

Daily temperature datasets used in tutorials on MachineLearningMastery.com was downloaded from Jason2BeownLee GitHub repository. This dataset contains average daily temperature from 01-Jan1981 to 31-dec-1990.

https://raw.githubusercontent.com/jbrownlee/Datasets/master/daily-max-temperatures.csv

## **APPLICATION OF AN APPROPRIATE BOX-JENKINS MODEL TO THE CHOSEN DATASET**

The Box-Jenkins model usually referred to as Box-Jenkins methodology is a popular time series forecasting technique that was developed in the early 1970s by George Box and Gwilym Jenkins.

The Box-Jenkins methodology consists of three main steps: Identification, Estimation and Diagnostic checking.

The identification stage uses the cross-correlation function between the output and input after they have both been transformed using the filter (i.e., ARMA model) that reduces xt to white noise, which is known as prewhitening. Estimation and diagnostic checking use extensions of their univariate counterparts, although these are not necessarily straight- forward. There have also been several other identification techniques proposed over the years, most notably based on prewhitening the input and output using individual filters (ARMA models). (Terence C. Mills,2015)

# EXPLORATORY DATA ANALYSIS (EDA)

Exploratory Data Analysis (EDA) is a useful method generally employed to understand data set by summarizing main characteristics of the data set and often visualizing by plotting.

Making informative visualizations (sometimes called plots) is one of the most important tasks in data analysis. It may be a part of the exploratory process-for example, to help identify outliers or needed for data transformations, or as a way of generating ideas for models. (Wes McKinney, 2017)

There are four commands which are used for Basic data exploration in Python.

head() : This helps to see a few sample rows of the data

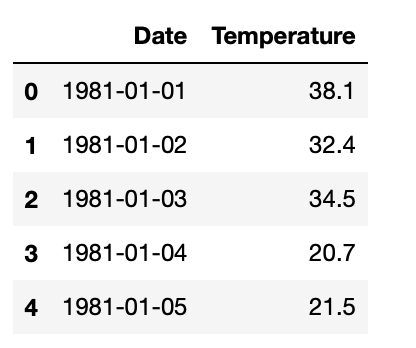
info() : This provides the summarized information of the data

describe() : This provides the descriptive statistical details of the data

nunique(): This helps us to identify if a column is categorical or continuous

Basic data exploratory analysis shows the dataset contains 2 columns of variables and 3650 rows of observations with date stamp and numerical datapoints. There are no missing values in all columns and rows are not duplicated in the dataset.

Fig 1: Dataset head and tail

**** **A table with numbers and a number of times

Description automatically generated with medium confidence**

In the time series Dataset, the date starts from 01-01-1980 and ends in 31-12-1990 as called up by the above head and tail functions.

Below line chart is used to show the temperature trends and how the pattern changes over time.

Fig 2: Temperature change over time series.

**A blue line graph with text

Description automatically generated with medium confidence**

# THE AUGMENTED DICKEY-FULLER TEST

Test for determining whether a process is stationary are called hypothesis test. The Augmented Dickey-Fuller (ADF) test is the most commonly used metric to assess a time series for stationary problems. This test posits a null hypothesis that a unit root is present in a time series for stationary problems. Depending on the results of the test, this null hypothesis can be rejected for a specific significance level, meaning the presence of a unit root test can be rejected at a significance level (Aileen Nielsen, 2020)

The ADF is applied on my time series dataset to determine whether the dataset is stationary or not.

Fig 3: ADF Result

A close-up of numbers

Description automatically generated

## **RESULT INTERPRETATION**

Test Statistic (-4.603703028446947): This is compared to the critical values to determine whether to reject the null hypothesis. It is a representation of the test statistic calculated by the ADF test.

P-value (0.00012715478116454037): This is the p-value associated with the test statistic. The p-value is compared to the chosen significance level (threshold defined according to the problem) to decide whether to reject the null hypothesis. The null hypothesis is typically rejected if the p-value is less than or equal to the significance level.

Lags Used (19): The number of lags used in the regression when computing the test statistic.

Number of Observations Used (3630): The total number of observations used in the analysis.

Critical Values: The dictionary contains critical values for the test statistic at different significance levels (1%, 5%, and 10%). These critical values are used to compare with the test statistic to determine whether to reject the null hypothesis.

Maximized Information Criterion (20156.23196850843): This is the maximized log-likelihood value.

Assuming the threshold for significance level is defined as 0.01, result for the p-value 0.00012715478116454037, suggests that there is enough evidence to conclude that the time series is stationary.

# TIME SERIES MODELLING.

The dataset index was split at 80% train set and 20% test set from the total number of rows and datapoints converted to an integer data type.

2920 rows and 2 columns are used for training while 730 rows and 2 columns are used for testing.

ADF was used to determine the P values of train split sample for the target variable-Temperature.

Diff (1) function was applied on the train set which calculates the difference between consecutive elements in the target variable. This is a common method for transforming a time series to make it stationary.

P value is 0.0 was achieved for the new column ‘Diff’ that was created by assigning it the values of the first difference of the "Temperature" column in the training data.

Fig 4: Graph of the train split samples

A blue sound wave graph

Description automatically generated

## **AUTOCORRELATION**

Autocorrelation is a measure of the correlation between a time series and its lagged values. Lag value is a value of a variable at a previous point in time.

Fig 5: Autocorrelation Plots of the column Temperature

A graph with blue lines

Description automatically generated

Fig 6: Autocorrelation Plots of the column Diff

A graph with blue lines

Description automatically generated

## **PARTIAL AUTOCORRELATION**

The partial autocorrelation function (PACF) is a tool for identifying the order of an autoregressive (AR) model.

The method parameter is set to "ywm" (Yule-Walker method), which is one of the methods used to estimate the partial autocorrelation.

Fig 7: Partial Autocorrelation Plots of the column Temperature

A graph with blue lines

Description automatically generated

Fig 8: Partial Autocorrelation Plots of the column Diff

A graph with blue lines

Description automatically generated

## **ARIMA MODEL**

ARIMA (AutoRegressive Integrated Moving Average) model is a popular time series forecasting method that combines order of autoregression-AR (p), differencing needed to achieve stationarity (d), and order of moving averages-MA (q)

At train and test split of 80% and 20% respectively for the target variable ‘Temperature’, I applied ARIMA model.

I used auto\_arima function to automatically select the best ARIMA model for training data. This function performs a grid search over possible combinations of p, d, and q parameters to find the model with the lowest AIC (Akaike Information Criterion) values for different orders (p, d, q).

The AIC is a measure of the model's goodness of fit, considering both the likelihood of the model and the number of parameters.

Fig 9: Model Result Summary

A screenshot of a computer screen

Description automatically generated

The above figure is a comprehensive summary of the SARIMAX (seasonal AutoRegressive Integrated Moving Average with eXogenous factors) model and its performance on the dataset.

The Ljung-Box test result of a low Q-value (0.01) suggests there is no significant correlation in the residuals at lag 1, which is a good sign for model performance.

The Jarque-Bera test indicates that the residuals are not normally distributed, possibly indicating a departure from normality.

The heteroskedasticity test for whether the variance of the error is constant over time suggests evidence of heteroskedasticity in the residuals.

### **FORECAST**

Forecasted values and confidence intervals for the next 10 temperature periods is predicted by extracting the last 10 observations from my test set. The forecast error was calculated by subtracting the forecasted values from the observed values.

Finally, the results are displayed in a DataFrame for better readability. This way, it is easy to see the observed values, forecasted values, and the corresponding errors for the last 10 observations in the test set.

Fig 10: Forecasted Results

A screenshot of a computer

Description automatically generated

## **SARIMAX**

I directly applied SARIMAX (seasonal AutoRegressive Integrated Moving Average with eXogenous factors) model on the dataset by defining the order of autoregression-AR (p), differencing needed to achieve stationarity (d), and order of moving averages-MA (q) as 2,1,2 respectively.

Fig 11: Visualize the performance and diagnostics of your SARIMAX model.

A graph and diagram of a graph

Description automatically generated

A graph of a graph with a red line

Description automatically generated with medium confidence

Standardized Residuals over Time: This plot helps to check for any patterns or trends in the residuals. The residuals should be randomly scattered around zero.

Histogram plus Estimated Density: This plot provides a histogram of the residuals along with an estimated density function. It helps assess the normality of the residuals.

Normal Q-Q (Quantile-Quantile) Plot: This plot compares the distribution of the standardized residuals to a theoretical normal distribution. Points generally fall along a straight line in the distribution.

Correlogram (ACF) of the Residuals: This plot shows the autocorrelation function (ACF) of the residuals, helping to identify any remaining patterns in the residuals.

Fig 12: SARIMAX Forecast vs Observed.

A graph showing a purple line

Description automatically generated

The plot visually assess how well SARIMAX model's forecast aligns with the actual observations. The confidence interval provides a sense of the uncertainty associated with the predictions.

I measured the R-squared score which is how well the predicted values match the actual values. A negative value of -144.68 indicates that my model is performing worse than a simple horizontal line (mean of the dependent variable). This could happen when the model's predictions are not capturing the variance in the dependent variable and are, on average, further from the actual values than a simple mean.

Furthermore, I measured the mean squared error. This is another common metric used to evaluate the performance of regression models.

The mean squared error is a measure of the average squared difference between the predicted values and the actual values. Lower MSE values indicate better model performance, while higher values indicate a larger average squared difference between predicted and actual values.

For my model a mean square error of 42.65 was obtained indicating a larger average squared difference between predicted and actual values.

# TEXT ANALYTICS

Text analytics using machine learning is a powerful tool for extracting insights from textual data, thereby automating tasks that would be impractical to perform manually and enabling better decision making in various domains.

The Best practice for handling text is the ‘Unicode sandwich’. This means that bytes should be decoded to ‘str’ as early as possible on input (e.g. when opening a file for reading). The meat of the sandwich is the business logic of your program, where text handling is done exclusively on str objects. You should never be encoding or decoding in the middle of other processing. On output, the str are encoded to bytes as late as possible. Pythons make it easy to follow the advice of the Unicode sandwich, because the open built-in does the necessary decoding when reading and encoding when writing files in text mode, so all you get from my\_file. read () and pass to my\_file-write(text) are str objects (Luciano Ramalho, 2015)

Fig 13: The Unicode sandwich

A close-up of a sandwich

Description automatically generated

The dataset from ‘train.csv’ focuses on real disasters, and the objective is to develop a predictive model that distinguishes between tweets related to actual disasters and those unrelated to disasters.

Fig 14: Data Visualization

A graph of a distribution of a product

Description automatically generated

## **MODELLING**

The dataset is cleaned and split into train and test sets of 80% and 20% respectively. CountVectorizer is used to convert text data into a bag-of-words representation and data was trained on Naive Bayes classifier model.

Fig 15: Naive Bayes classifier results

A number of numbers on a white background

Description automatically generated

For the Naive Bayes classifier, an accuracy score of 78% was achieved and below confusion matrix plot shows the predicted class and the actual classes.

Fig 16: Confusion Matrix model.

Diagram

Description automatically generated

Fug 17: Confusion Matrix for Naive Bayes classifier model

A yellow green and purple squares with numbers

Description automatically generated

From the confusion matrix table, 658 (True Positive) correctly predicted as unrelated to disasters.

536 (True Negative) correctly predicted as related to disasters from the dataset.

173 (False Positive) incorrectly predicted as unrelated to disasters while 156 (False Negative) were incorrectly classified as related to disasters from the dataset.

**DASHBOARD**

I represented on an interactive dashboard the temperature dataset used for time series model predictions. The aim is to visually display all the information presented in the dataset in a consolidated and easy to understand format.

Prior to creating the dashboard, the design of the proposed dashboard was created using draw.io.

Fig18: Wireframe for dashboard

A screenshot of a graph

Description automatically generated

From the proposed wireframe, the dashboard will include line chart, Histogram, bar plots and text.

## **ELEMETS OF DASHBOARD**

Fig 19: Interactive Date vs Temperature line chart

A graph showing the time of a graph

Description automatically generated with medium confidence

Fig 20: Interactive scatter plot with a range slider for filtering by time period

A graph showing the time of a graph

Description automatically generated with medium confidence

Fig 21: Interactive statistical description of the dataset.

A screenshot of a graph

Description automatically generated

Fig 22: Average Temperature per year

A blue rectangular bars with white text

Description automatically generated

Fig 23: Histogram with average, median and Rayleigh fit.

A graph with a blue line

Description automatically generated

Fig24: Text with Mean Square Error score

A green numbers on a white background

Description automatically generated

**REFERENCES**

Aileen Nielsen, 2020 (Practical Time Series Analysis, Prediction with Statistics & Machine Learning.Pg.1, Pg.2, Pg.84)

Terence C. Mills, 2019 (Applied Time Series Analysis, A Practical guide to Modelling and Forecasting. Pg.203)

https://raw.githubusercontent.com/jbrownlee/Datasets/master/daily-max-temperatures.csv

Wes McKinney, 2017 (Python for Data Analysis. Data Wrangling with Pandas, NumPy, and IPython. Second edition. Pg.19)

Luciano Ramalho, 2015 (Fluent Python, Clear, concise, and effective Programming. Pg, 111)