TABA: Time Series Analysis and Logistic Regression

Yash Bhargava
Dept. of Computing
National College of Ireland
Dublin, Ireland
x22220861@student.ncirl.ie

***Abstract*—This study consists of two parts the first part was about time series analysis and the second part was about Logistic Regression. In Part A of this study, I have done the time series analysis of the weather dataset in which I have done analysis on the 'maxtp' column and applied 6 models of time series in which I have discovered SARIMA model outperformed all the other time series model and worked the best with this dataset. With this SARIMA model, I have forecasted the values on the test set in which we have 10 months of data. In the second Part B, I analysed cardiac data in the dataset consisting of 100 patients in which I had to predict the cardiac condition i.e., present or absent based on the factors, applied Logistic Regression, and got an accuracy of 83% using Python on Jupiter Notebook.**

***Index Terms*—Time Series, Exponential Smoothing, ARIMA, SARIMA, Logistic Regression**

I. Part-A Time Series Analysis

Dataset 1: Weather

A. Dataset Description and Data Understanding

The weather dataset consists of various variables and the data was of Ireland weather data recorded by Dublin Airport on which I have done time series analysis on variable i.e., 'maxtp' which was the Maximum Air Temperature in degrees C. The term time series refers to the data in which the data was chronologically ordered based on the time and has some pattern in the data such as trend, seasonality and the granularity of data can be daily, weekly, monthly and yearly data as this was our daily data which have 29889 observations and 9 variables from 1st January 1942 to 31st October 2023, so it was a huge data but I have the time series analysis on dataset from 2019 to 2022 and forecasted the values of 2023. So, I have split the dataset from 2019 to 2022 into the training set and from 2023 as a testing set. So, in our training set, we have 1461 observations and 1 variable which was the 'maxtp' column on which I have performed time series analysis, and the testing set has 304 observations and 1 variable. To see the pattern of the data and if there was any trend and seasonality in the data, I plotted the training data on the graph, and we can see that there was no cyclic nature and trend in the graph but there was seasonality found in the graph as it can be seen in Figure 1.

B. Exploratory Data Analysis

In this part of the analysis, we have converted the data type of the date as it was in object data type then I set the date column as an index and have done subsetting on the dataset and split the dataset into train and test set based on the time i.e., from 2019 to 2022 as a training set and from 2023 as testing set. Before the decomposition of the training data, I plotted the data as can be seen in Figure 1 and we can see that the data has seasonality as the maximum air temperature was high in July as these are considered the summer months but there was no trend found in the data as the maximum air temperature remains constant. Then I did the decomposition of the training data which separately shows us the trend, seasonality, and the irregular component. In the analysis of time series, we have two types of decomposition of data one is additive, and the other is multiplicative. So, in our case, I have considered the multiplicative decomposition in which our equation can be shown as:

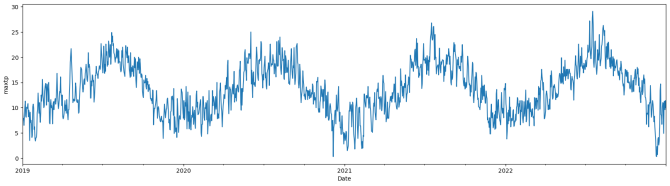

Fig. 1. Plotting the training set

$$Y(t) = T(t) \times S(t) \times I(t) \quad (1)$$

In which our Y(t) was the observed time series, T(t) was the trend component, S(t) was the seasonal component, and I(t) was the irregular component and the value of I(t) in the multiplicative model was always near to 1 as in the equation it was multiplied with the trend and seasonal components. The decomposition of our multiplicative model can be seen in Figure 2.

C. Model Building

In this section, I have applied 7 models which consist of 2 simple time series i.e., simple moving average and weighted moving average, 3 exponential smoothing models i.e., simple exponential smoothing, Holt's Linear Trend model and Holt's Winters exponential smoothing model, Auto Regressive Integrated Moving Average (ARIMA) model and Seasonal Auto-Regressive Integrated Moving Average (SARIMA) as

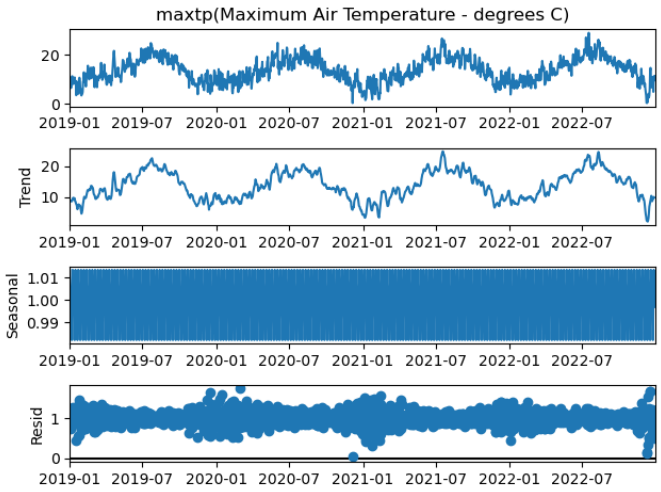

Fig. 2. Decomposition of training set for Time series

SARIMA was applied only when there was seasonality in the data.

1) Model 1: Simple Moving Average (SMA): The moving average model was the simplest of all the time series models as the prediction was based on the average of past values i.e., in our simple moving average model we have predicted the values of the train data set based upon the rolling window sizes as 2 and 3 and fitted these models on the training set plotted the fitted values of the simple moving average on the graph as we can see in the Figure 3.

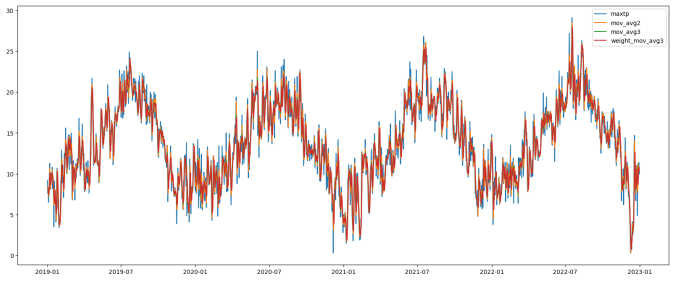

Fig. 3. Plotting the SMA and WMA

2) Model 2: Weighted Moving Average (WMA): Weighted Moving Average was also like moving average but in this case, we assign some weights to previous observations based on how close the previous value was from the current observation we will be predicting. In our case, I have calculated the weighted moving average based on the weights like 0.4 was assigned to the preceding observation, 0.35 to its previous and weights were reduced as per the previous observations and taken the average as the prediction of the current observation and fitted the values in the training set and it can be seen in Figure 3.

3) Model 3: Simple Exponential Smoothing Model (SES): Simple Exponential Model is a type of Exponential Time Series smoothing model which was used to apply the single

SimpleExpSmoothing Model Results

Dep. Variable:	maxtp(Maximum Air Temperature - degrees C)	No. Observations:	1461
Model:	SimpleExpSmoothing	SSE	7364.124
Optimized:	True	AIC	2367.166
Trend:	None	BIC	2377.740
Seasonal:	None	AICC	2367.193
Seasonal Periods:	None	Date:	Mon, 01 Jan 2024
Box-Cox:	False	Time:	00:47:59
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.6626912	alpha	True
initial_level	8.6729135	l.0	True

Fig. 4. Summary of Simple Exponential Smoothing model

variable with no trend and seasonality as it was a baselined smoothing model. When I fitted this model on the training it got an Akaike's information criteria (AIC) value of 2367.166 in the summary of this model in Figure 4 and forecasted the values on testing set and got MAE of 5.63, MAPE of 37% and RMSE of 5.63 and the potted the forecasted value on the graph as it can be seen in Figure 5.

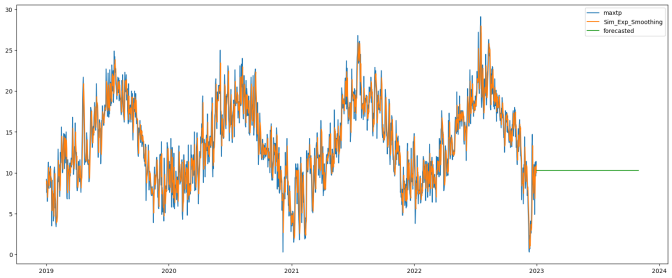

Fig. 5. Plot for forecasting using SES

4) Model 4: Holt's Linear Trend Exponential Smoothing Model: Holt's Linear Trend model also known as the double exponential smoothing model, was used when there was a trend pattern but no seasonality in the data. In our case, when I fitted this model on our training set and got an AIC value of 2371.165 in the summary of this model in Figure 6 and

Holt Model Results

Dep. Variable:	maxtp(Maximum Air Temperature - degrees C)	No. Observations:	1461
Model:	Holt	SSE	7364.120
Optimized:	True	AIC	2371.165
Trend:	Additive	BIC	2392.313
Seasonal:	None	AICC	2371.223
Seasonal Periods:	None	Date:	Mon, 01 Jan 2024
Box-Cox:	False	Time:	00:47:59
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.6626882	alpha	True
smoothing_trend	0.000000	beta	True
initial_level	8.6713681	l.0	True
initial_trend	0.0011227	b.0	True

Fig. 6. Summary of Holt's Linear Trend Smoothing model

forecasted the values based on this model for the testing set and plotted the graph for the forecasted values it can be seen that this model did forecast the values but it did not fit the data well and predicted the average of the values based on the past observations in Figure 7.

5) Model 5: Holt's Winters Exponential Smoothing Model: This is a type of Exponential Smoothing Model which was considered when there was a trend along with the seasonal patterns which repeat after regular intervals this model better fits our training as compared to the previous two exponential

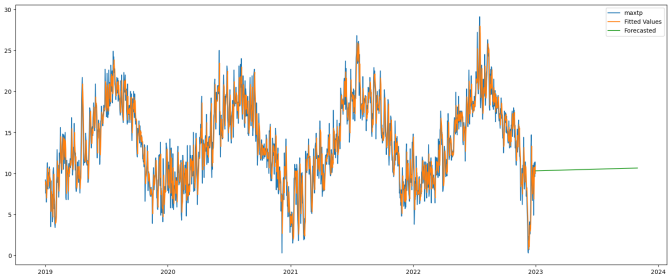

Fig. 7. Plot of Holt's Linear Trend Smoothing model

smoothing models as it got lesser RMSE value of 4.63 on forecasting the test set values but the AIC value of 2642.554 which was highest among the other exponential models as seen in the summary of this mode in Figure 8 and have plotted the

ExponentialSmoothing Model Results

Dep. Variable:	maxtp(Maximum Air Temperature - degrees C)	No. Observations:	1461
Model:	ExponentialSmoothing	SSE	5848.714
Optimized:	True	AIC	2642.554
Trend:	Additive	BIC	4270.912
Seasonal:	Multiplicative	AICC	2810.224
Seasonal Periods:	304	Date:	Mon, 01 Jan 2024
Box-Cox:	False	Time:	00:48:11
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.6584863	alpha	True
smoothing_trend	0.0005246	beta	True
smoothing_seasonal	7.3136e-05	gamma	True
initial_level	10.933446	l.0	True
initial_trend	0.0042601	b.0	True
initial_seasons.0	1.1130335	s.0	True

Fig. 8. Summary of Holt's Winters Exponential Smoothing model

graph for the forecasted values in the test sets and it can be seen that the model was not fitted well on the training data as compared to other exponential models in Figure 9.

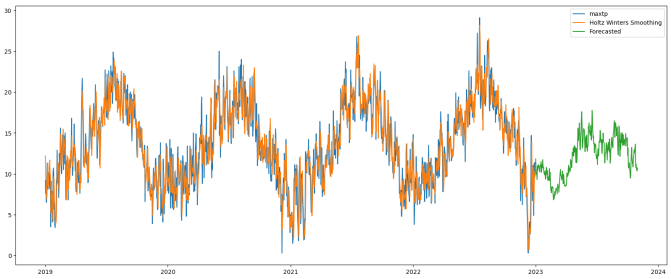

Fig. 9. Plot of Holt's Winters Exponential Smoothing model

6) Model 6: Auto Regressive Integrated Moving Average (ARIMA) Model: ARIMA model consists of 3 parts AR+I+MA which AR stands for Auto-Regressive which constructs the value of p in the non-seasonal order of ARIMA, I stands for Integrated which means the number of times the data was differenced for the data to be stationary, and MA stands for Moving Average which constitutes the value of q in the non-seasonal order of ARIMA model. But firstly, before the ARIMA model is applied, there are some prerequisites for the data that need to be taken care of and the data needs to be stationary. So, to check the stationarity of our training set we have used the Augmented Dickey-Fuller test which was the most common test. After testing the stationarity of the data, we have the p-value of the training set as 0.08 which can be seen in Figure 10, which was greater than

```
Test Statistic                -2.663824
p-value                        0.080510
Lags Used                     17.000000
Number of Observations      1443.000000
dtype: float64
```

Fig. 10. Dickey- fuller test results on train data

the threshold value of significance which was 0.05. So, for converting the non-stationary data into stationary data I have done differencing of the data and after the differencing I have again tested it with the Dickey Fuller test and now got the p-value as 6.316923881482345e-25 which was below the threshold value of 0.05 which can be seen in Figure 11. So,

```
Test Statistic               -1.332610e+01
p-value                       6.316924e-25
Lags Used                     1.600000e+01
Number of Observations        1.443000e+03
dtype: float64
```

Fig. 11. Dickey- fuller test results after first differencing on train data

after the transformation of the variable, I have plotted the training set data on a graph as can be seen in Figure 12,

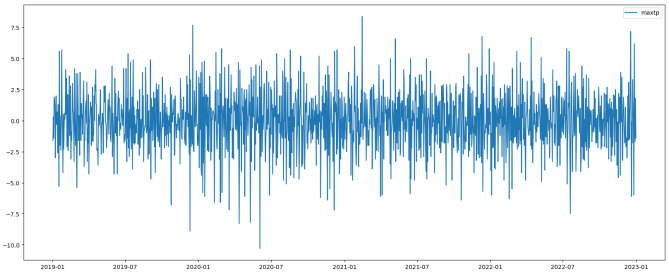

Fig. 12. Plot for training set after transformation

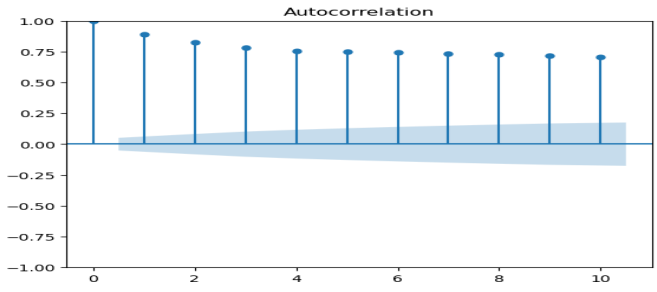

Fig. 13. ACF plot for ARIMA

I have plotted Autocorrelation and Partial Autocorrelation plots also known as ACF and PACF plots in which I have taken 10 lags. The reason for plotting the ACF plot was to determine

the value of q and got different values of q and the PACF plot was used to determine the values of p as can be seen in Figure 13 and Figure 14. After plotting, I tried closer combinations

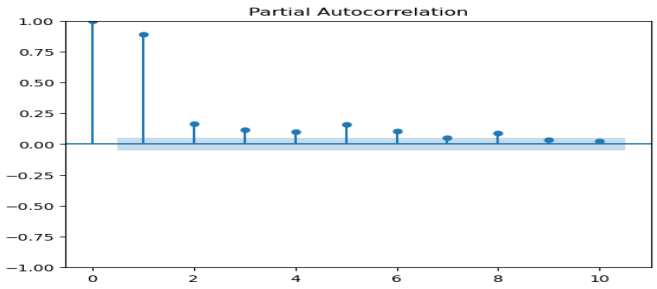

Fig. 14. PACF plot for ARIMA

of orders of ARIMA like, I have built 5 ARIMA models with different combinations in which the ARIMA model having the non-seasonal order (5,1,9) outperformed all the other ARIMA models with closer combinations with lowest AIC value of 6380.22 in the summary of this model in Figure 15 and then I have forecasted with the test set and plotted the graph as it can be seen in Figure 16

SARIMAX Results

Dep. Variable:	maxtp(Maximum Air Temperature - degrees C)	No. Observations:	1461
Model:	ARIMA(5, 1, 9)	Log Likelihood	-3175.111
Date:	Mon, 01 Jan 2024	AIC	6380.223
Time:	16:46:05	BIC	6459.516
Sample:	01-01-2019	HQIC	6409.802
	- 12-31-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.2369	0.079	-2.987	0.003	-0.392	-0.081
ar.L2	0.2085	0.068	3.077	0.002	0.076	0.341
ar.L3	-0.3159	0.061	-5.145	0.000	-0.436	-0.196
ar.L4	0.3223	0.067	4.846	0.000	0.192	0.453
ar.L5	0.8621	0.076	11.361	0.000	0.713	1.011
ma.L1	-0.1074	0.083	-1.295	0.195	-0.270	0.055
ma.L2	-0.4629	0.087	-5.312	0.000	-0.634	-0.292
ma.L3	0.2308	0.071	3.249	0.001	0.092	0.370
ma.L4	-0.5627	0.070	-8.044	0.000	-0.700	-0.426
ma.L5	-0.8520	0.080	-10.697	0.000	-1.008	-0.696
ma.L6	0.3589	0.038	9.454	0.000	0.285	0.433
ma.L7	0.1316	0.034	3.845	0.000	0.065	0.199
ma.L8	0.1484	0.028	5.330	0.000	0.094	0.203
ma.L9	0.1679	0.028	5.977	0.000	0.113	0.223
sigma2	4.5371	0.153	29.627	0.000	4.237	4.837

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	21.87
Prob(Q):	0.98	Prob(JB):	0.00
Heteroskedasticity (H):	0.99	Skew:	0.01
Prob(H) (two-sided):	0.94	Kurtosis:	3.60

Fig. 15. Model Summary for ARIMA (5,1,9)

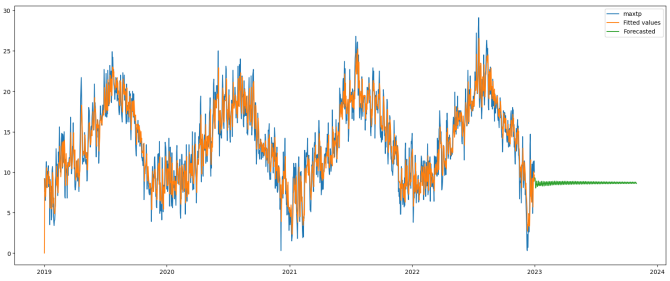

Fig. 16. Forecast Plot for ARIMA (5,1,9) model

and calculated the mean absolute error (MAE), mean absolute percentage error (MAPE), and Root Mean Squared

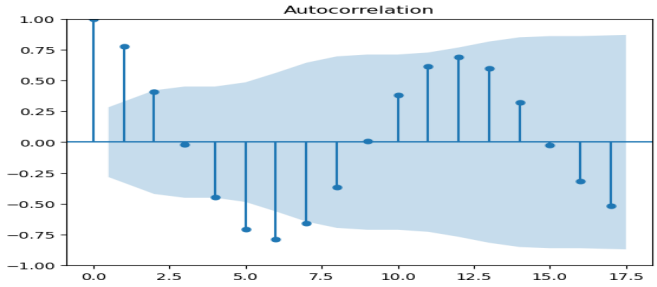

Fig. 17. ACF plot for SARIMA

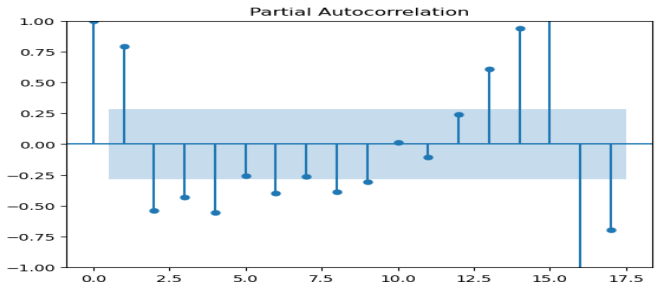

Fig. 18. PACF plot for SARIMA

Error (RMSE) which was found to be 6.30, 0.43 and 7.94 respectively.

7) Model 7: Seasonal Auto-Regressive Integrated Moving Average (SARIMA) model: SARIMA was an extension of the ARIMA model in which it overcomes the disadvantages of the ARIMA model as it was generally considered when there was a seasonal pattern in the data as it takes an additional parameter as seasonal order in the SARIMAX function in Python seasonal order parameter was represented as (P, D, Q, m) in which P was represented as autoregressive part for the seasonal component, D stands for differencing for the seasonal component and Q represents the moving average part and the value of P, Q was determined by using the ACF and PACF plots plotted for the SARIMA model using the month-wise data of training set as it can be seen in Figure 17 and Figure 18.

SARIMAX Results

Dep. Variable:	maxtp(Maximum Air Temperature - degrees C)	No. Observations:	48
Model:	SARIMAX(1, 1, 2)x(0, 1, [1], 12)	Log Likelihood	-58.363
Date:	Mon, 01 Jan 2024	AIC	126.726
Time:	16:46:21	BIC	134.503
Sample:	01-31-2019	HQIC	129.411
	- 12-31-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.4771	0.257	-1.859	0.063	-0.980	0.026
ma.L1	0.0809	40.723	0.002	0.998	-79.735	79.897
ma.L2	-0.9184	37.309	-0.025	0.980	-74.042	72.205
ma.S.L12	-0.9953	57.877	-0.017	0.986	-114.431	112.441
sigma2	0.8916	76.271	0.012	0.991	-148.596	150.379

Ljung-Box (L1) (Q):	0.54	Jarque-Bera (JB):	1.50
Prob(Q):	0.46	Prob(JB):	0.47
Heteroskedasticity (H):	0.74	Skew:	-0.26
Prob(H) (two-sided):	0.62	Kurtosis:	2.13

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

Fig. 19. Summary for SARIMA (1,1,2) (0,1,1,12)

After determining the seasonal order I have chosen the value

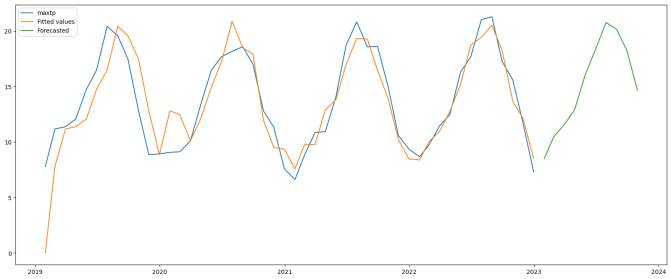

Fig. 20. Forecasted test set values plot for best SARIMA (1,1,2) (0,1,1,12)

of m in seasonal order as 12 as the data was converted to month-wise data for applying the SARIMA model and tried various closer combinations of seasonal order and found that the SARIMA model with non-seasonal order as (1,1,2) and seasonal order as (0,1,1,12) has performed the best among the 3 SARIMA models applied with lowest AIC value of 126.726 in the summary of the model as it can be seen in Figure 19 and mean absolute error (MAE) of 0.75, mean absolute percentage error of 0.04 and root mean squared error of 0.96. Finally, I have forecasted the test set values for the best SARIMA model and plotted the graph as can be seen in Figure 20.

D. Summary

In the time series analysis of weather data, I applied a total of 7 models, and it was found that the SARIMA (1,1,2) (0,1,1,12) model performed the best out of all the 7 models with the lowest AIC value of 126.726 and mean absolute error (MAE) of 0.75, mean absolute percentage error of 0.04 and root mean squared error of 0.96 was the best among all the models applied.

II. Part-B Logistic Regression

Dataset 2: Cardiac

A. Data Description and Understanding

Logistic Regression is a classification-based statistical model that is used in the prediction of a categorical variable and it is the most commonly used binary-classification model in which there is a dependent variable, which is denoted as the y variable, and two or more variables that are independent of each other, denoted as the x variable, as logistic regression assumes that there was a linear relationship between dependent and independent variables. The equation of the logistic regression is shown as:

$$P(Y=1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n)}} \quad (2)$$

The dataset I used for this study was cardiac consisting of 100 observations and 5 variables in which the target variable was 'cardiac_conditon' for which the likelihood of the variable was (Present/Absent) as it was a binary class classification problem. The features or the independent variables in this dataset were 'caseno' which was a unique ID of the patient, 'age' refers to the age of the patient, the weight represents the weight of the patient in kilograms, 'gender' represents the gender of the patient that whether the patient was Male or Female, and 'fitness_score' refers to the measure of fitness out of 100 which were responsible for predicting the 'cardiac_condition' of the patient. For this dataset, I have used Logistic Regression using the sklearn package in Python.

B. Descriptive Statistics and Exploratory Data Analysis

In this step, we did some analysis and understood the data using the info() method of the pandas library by which we got to know that there are no missing values in the data, then using the describe function of the pandas library we got the descriptive statistics of all the continuous columns i.e., 'caseno', 'age', 'weight' and the 'fitness_score' and found out the count which represents the number of observations in the data, mean which shows the average for each variable, std that represents the standard deviation for each variable, min refers to the minimum value for each variable, 25% represents the 25th percentile value of each variable, 50% represents the median for each of variables, 75% represents the 75th percentile for each variable and the max represents the maximum values for each variable, it was also known as five-point summary tables which were shown in Figure 21. Before the model-building phase, we also checked for outliers

	count	mean	std	min	25%	50%	75%	max
caseno	100.0	50.5000	29.011492	1.00	25.7500	50.50	75.2500	100.00
age	100.0	41.1000	9.142530	30.00	34.0000	39.00	45.2500	74.00
weight	100.0	79.6603	15.089842	50.00	69.7325	79.24	89.9125	115.42
fitness_score	100.0	43.6298	8.571306	27.35	36.5950	42.73	49.2650	62.50

Fig. 21. Descriptive Statistics

by plotting the boxplot using the Seaborn library in Python and found out that there are very few outliers in the 'age' column as shown in Figure 22, so we have not treated the

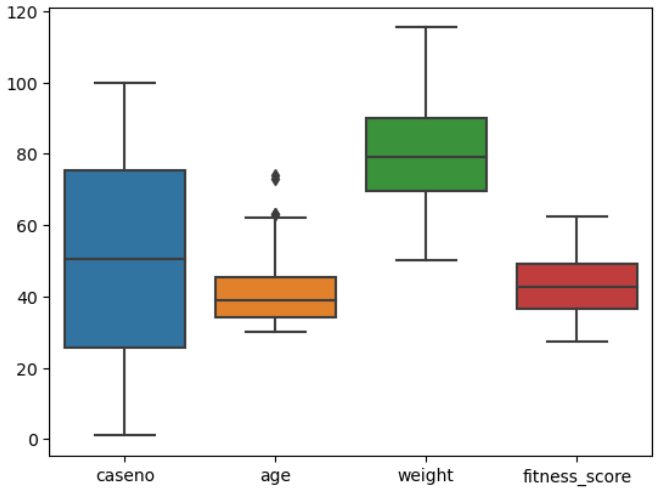

Fig. 22. Boxplot for Outliers

outliers as the size of the data was small. So, it will not

affect the model's performance very much. Then, I checked the correlation between the features using the heatmap function of the seaborn library in Python as it was shown in Figure 23. Finally, I have encoded the categorical variables i.e., 'gender'

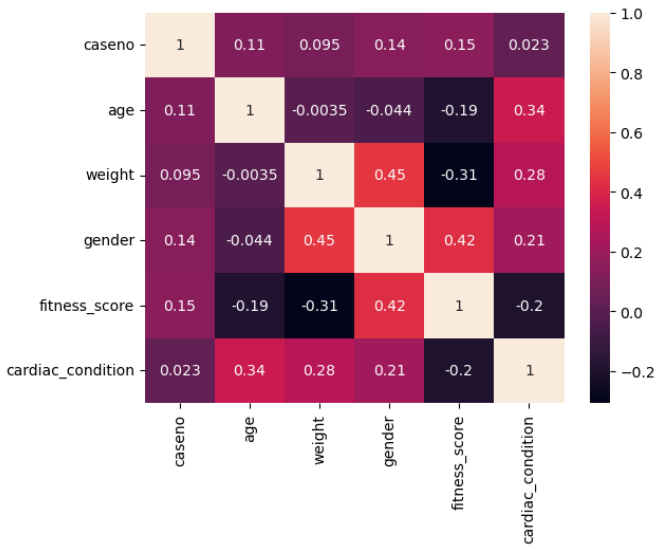

Fig. 23. Correlation Heatmap

and the target variable 'cardiac_condition'. Then I discovered that caseno variable was insignificant to be used in the model-building phase.

C. Model Building

In this phase, I have first split the data into two parts i.e., training set and testing set for which 70% of the data goes into training set and the 30% data goes to the test set. The Logistic Regression was built using the sklearn library and the model was fitted on the train data after the model was trained with training data, now we predicted the cardiac condition on the test data and calculated the evaluation metrics on which the model was judged.

D. Evaluation Metrics

After the values were predicted on the test data, we made the confusion matrix as shown in Figure 24, and found that 25

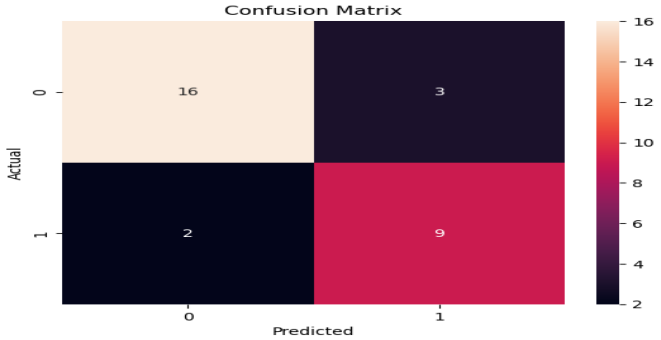

Fig. 24. Confusion Matrix

	precision	recall	f1-score	support
0	0.89	0.84	0.86	19
1	0.75	0.82	0.78	11
accuracy			0.83	30
macro avg	0.82	0.83	0.82	30
weighted avg	0.84	0.83	0.83	30

Fig. 25. Classification Report

observations were correctly predicted out of 30 observations, The classification report is shown in Figure 25 and found that the accuracy of 83%, Precision of 0.75, Recall of 0.82, F1_Score of 0.78 and ROC_AUC score of 0.83 which shows that it was the appropriate model for this dataset.

III. Summary

In this final section, we conclude that in the first analysis i.e., Time Series Analysis of the weather dataset we have applied a total of 7 different time series models which 2 simple time series models i.e. Simple Moving Average and Weighted Moving Average, 3 models of Exponential Smoothing models, tried different close variations of ARIMA models, and SARIMA models and in the final review we can conclude that SARIMA performed the best among all the above-performed models because of the seasonal nature of the data with the lowest AIC value of 126.72 and RMSE of 0.96. In the second part of the analysis, we applied Logistic Regression and got a good accuracy of 83%, Precision of 0.75, Recall of 0.82, F1_Score of 0.78 and ROC_AUC score of 0.83 which shows that it was the appropriate model after performing all the preprocessing steps. Finally, in this study, we can have considered the time series analysis and Logistic Regression and were able to find the appropriate acceptable models for the datasets.