STATISTICS FOR DATA ANALYTICS
TABA
Time Series Analysis and Logistics Regression

Vipin Sharma
Dept. of Computing
National College of Ireland
Dublin, Ireland
x22207406@student.ncirl.ie

Abstract - The report is divided into two distinct parts (Part A and Part B). Part A focuses on the Time Series Analysis which is performed on the weather dataset and Part B focuses on the Logistic Regression which is also performed on the Cardiac dataset. Time Series performed a detailed analysis of the weather dataset using various models, Simple time series model, Simple Exponential Smoothing, Holt linear trend model (Double Exponential Smoothing), and ARIMA. After performing all the models, the author compared every model, it was discovered that ARIMA was the good model for this dataset with the lower RMSE value. In Part B, the author applied a logistic regression on the Cardiac dataset. Using logistic regression to determine the various factors that affect the cardiac condition and performing all the Statistical analyses on the dataset, the accuracy score of the logistic regression model comes to 0.73 and the balanced accuracy score is 0.67.

Index Terms: Time Series, Smoothing, ARIMA, Logistic Regression, Python

I. PART A - TIME SERIES ANALYSIS

Dataset 1 : Weather Dataset

A. Data Description and Understanding

Time Series analysis is one of the statistical techniques that deals with time-ordered data points. The dataset in the time series where the date is indexed in time and the time interval is equal. The data points collected over a period of time are included in the time series which also indicates the underlying pattern such as trend, seasonality, cyclic, and error, all underlying patterns used to evaluate the performance. Time series analysis is mostly used in stock forecasting, weather forecasting, the financial sector, and many more. It is used to predict new values based on the previous observed values. In this report, the author works with the weather dataset that is provided, the weather dataset contains 5 columns and 29889 rows and according to the descriptor, the author works on the cbl (mean CBL Pressure-hpa) column with the date column and date column which becomes the index of the weather dataset for the time series analysis. The weather dataset has 19 years of data from 1942 to 2023.

B. Data Analysis

The first step of Data Analysis is to check the datatypes and Null values using the Python function .info() and .isnull(), the author found that the data do not contain any null values but the date column datatypes is object type which must be a datetime datatype when perform the time series analysis so, first convert the datatype of the date column using the pandas library function pd.to_datetime(). It will convert our date column datatype in the datetime64[ns]. After this, the author set the date column as an index using the .set_index function. Now it is time to decompose the Time series by doing the decomposition we can see the Trend, seasonal components, and Irregularity components and the decomposition is the part of EDA to understand the data better. There are two types of decomposition in time series Multiplicative and Additive both show similar type of results the basic difference between them is Irregularity components when performing the multiplicative model the irregularity is close to 1 and when performing the additive model the irregularity is close to 0. For this dataset, the author uses the additive model and there is no trend and seasonality shown also the Irregularity values close to 0 shown in Figure 1 it means there is no problem go ahead for further analysis and create a model.

C. Model Building

In the Model Building section, the author developed various models (Simple Moving average), the simple Exponential Smoothing Model, the Hot Linear model (Double Exponential smoothing model), and ARIMA. The most accurate model will be selected as the best model which will be determined by the metrics. The author will also be performing forecasting using the previous weather data but before that author plot a simple graph of the data shown in figure 2.

Model 1: Simple Moving Average

When dealing with the time series model the simple moving average model is a very basic model which applies to the time

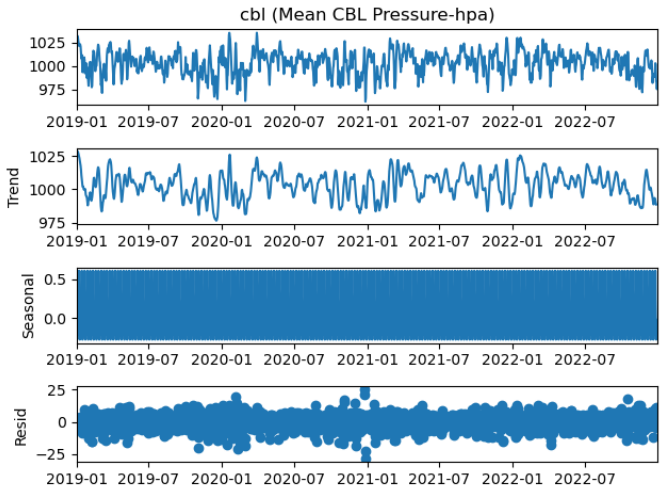

Fig. 1. Additive Decomposition Plot

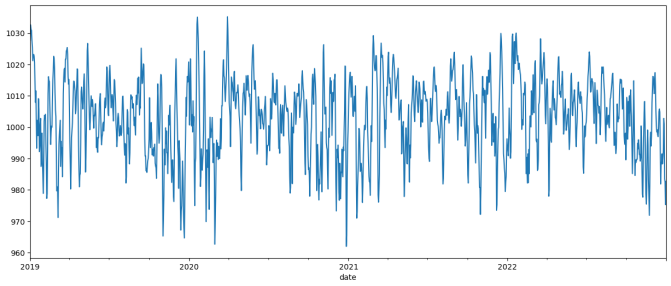

Fig. 2. Simple Time Series Plot

series. This model is used for smoothing time series data and this model is used to calculate the mean of the column values over a given window size. For this data, the author takes the window size 10 and plots the graph, the figure 3 shows the smoothing time series if compared to the original data.

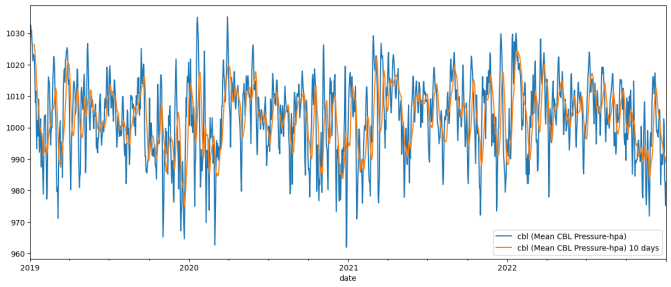

Fig. 3. Simple Moving Average Plot

Model 2: Simple Exponential Smoothing Model

The Exponential smoothing model gives the new predicted values based on the previous values and the past errors. The author applies the simple exponential smoothing model to the dataset and when fit the model by using the .fit function then it will give the summary of the model which shows the AIC value 5439.735 which is shown in Figure 4

SimpleExpSmoothing Model Results

Dep. Variable:	cbl (Mean CBL Pressure-hpa)	No. Observations:	1461
Model:	SimpleExpSmoothing	SSE	60320.911
Optimized:	True	AIC	5439.735
Trend:	None	BIC	5450.309
Seasonal:	None	AICC	5439.762
Seasonal Periods:	None	Date:	Sun, 31 Dec 2023
Box-Cox:	False	Time:	21:52:55
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	1.0000000	alpha	True
initial_level	1027.8000	l.0	True

Fig. 4. Summary of SimpleExpSmoothing Model

then performs the prediction and forecasting for the test data size which is shown in figure 5 . After performing the forecasting and the prediction the root mean square (RMSE) value is 23.945 and the Mean Absolute Error is 21.168.

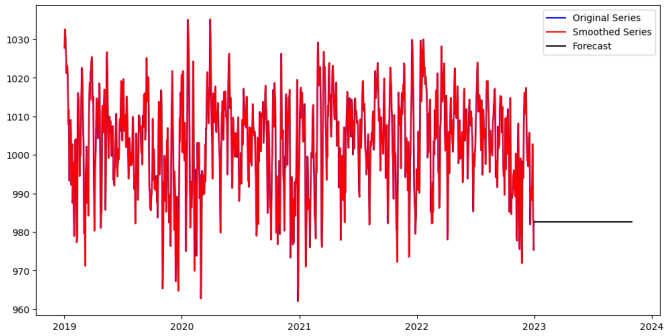

Fig. 5. Simple Exponential Smoothing Plot

Model 3: Holt-Linear Trend Model (Double Exponential Smoothing)

The Hot Linear Trend model is also known as the Double Exponential Smoothing model. When the author applies this model to the training dataset and prints the summary the AIC value comes is 5480.790 and the smoothing parameters apha = 0.994 for the level, beta=0.023 for the trend which is shown in Figure 6.

Holt Model Results

Dep. Variable:	cbl (Mean CBL Pressure-hpa)	No. Observations:	1461
Model:	Holt	SSE	61870.376
Optimized:	True	AIC	5480.790
Trend:	Additive	BIC	5501.937
Seasonal:	None	AICC	5480.847
Seasonal Periods:	None	Date:	Sun, 31 Dec 2023
Box-Cox:	False	Time:	21:52:55
Box-Cox Coeff.:	None		

	coeff	code	optimized
smoothing_level	0.9945140	alpha	True
smoothing_trend	0.0236111	beta	True
initial_level	1027.7032	l.0	True
initial_trend	-0.3564387	b.0	True

Fig. 6. Summary of Holt Model

When performing forecasting on the test data as shown in Figure 7, the forecasting shows a downward trend, and the matrices, root mean squared error is 77.284 and Mean absolute error is 71.545.

Model 4: ARIMA Model

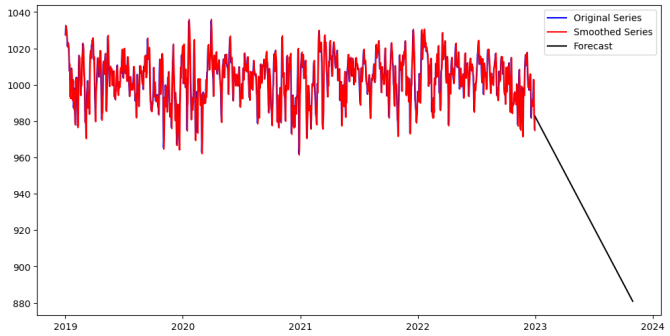

Fig. 7. Hot Linear Trend Plot

The Auto-Regressive Integrated Moving Average model is referred to as ARIMA. It is one of the best models to use for time series forecasting. The ARIMA takes three parameters (P, D, Q), P represents the auto-regressive which shows the correlation between the previous time and the current time and D represents the order of differentiation and Q represents the order of moving average. Before applying the ARIMA model first check whether the data is stationarity or not because the ARIMA model applies only to the stationary data. For this author used the Augmented Dickey-Fuller Test (ADFT), which gives the p-value and the p-value is less than the significant value(0.05) show in Figure 8 it means our dataset is stationary so there is no need for differencing.

```
Results of Dickey-Fuller Test:
Test statistic                -1.249242e+01
p-value                        2.925573e-23
#Lags Used                     2.000000e+00
Number of Observation Used     1.458000e+03
Crtical value (1%)            -3.434843e+00
Crtical value (5%)            -2.863524e+00
Crtical value (10%)           -2.567826e+00
dtype: float64
Dataset is stationary.
```

Fig. 8. Augmented Dickey-Fuller Result

After this author finds the P and Q values by plotting the autocorrelation function (acf) and partially auto-correlation function (pacf) plot which are shown in Figures 9 and 10. Using the acf and pacf values author creates four models ARIMA(2,0,1), ARIMA(1,0,2), ARIMA(3,0,1), ARIMA(1,0,3) compares all the models, and concludes that model2 ARIMA(2,0,1) shows the lowest AIC value, also by applying the auto_arima function which comes from pmdarima module gives the ARIMA(2,0,2) if compares the ARIMA(2,0,1) and ARIMA(2,0,2) there is very less difference between the AIC value. The summary table of ARIMA(2,0,1) show in figure 11 and ARIMA(2,0,2) show in figure 12

After this figure 13 shows the plot of residual values and Using the Ljung-Box author checks whether there is significant autocorrelation in the residuals of a time series model at various lags. It shows that the all the lags up to 10 the p value is

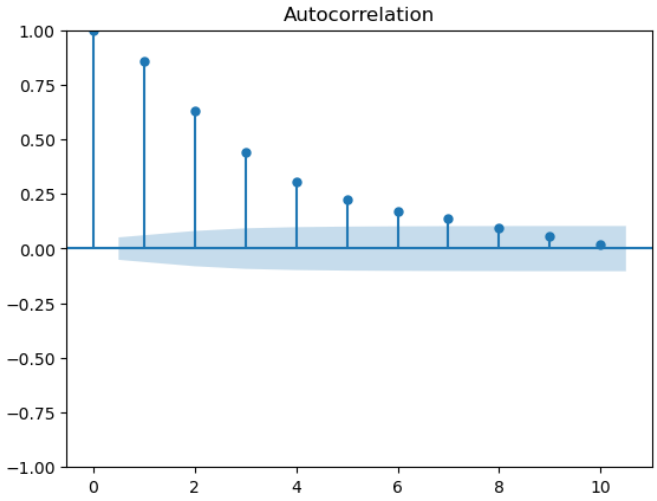

Fig. 9. Auto-correaltion Plot

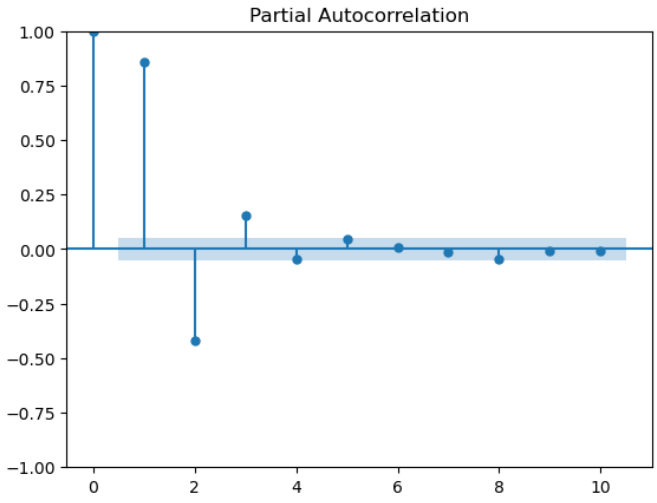

Fig. 10. Partial Auto-correlation Plot

not significant which indicates that there is no serial correlation present. For the evaluation, we plot a residual which shows a normal distribution graph in Figure 14

which means the evaluation is going in the right direction. At the end author performs a forecasting which is shown in Figure 15 and calculates the evaluation matrices, RMSE is 5.596, MAE is 4.275 and MAPE is 0.004 on the train when we fit the model, and when performing prediction on the test data, the RMSE value is 12.542, MAE is 9.871 and MAPE is 0.010.

SARIMA Model

The Seasonal AutoRegressive Integrated Moving Average is referred to as SARIMA, it is the extension of the ARIMA model. With an extra element of seasonality in the time series data and represents the seasonal component of the model. Depending on the data seasonality can occur weekly, monthly, in winters and summers, or at any other fixed interval. In

SARIMAX Results

Dep. Variable:	cbl (Mean CBL Pressure-hpa)	**No. Observations:**	1461
Model:	ARIMA(2, 0, 1)	**Log Likelihood**	-4581.827
Date:	Sun, 31 Dec 2023	**AIC**	9173.653
Time:	21:21:47	**BIC**	9200.088
Sample:	01-01-2019	**HQIC**	9183.514
	- 12-31-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1003.1762	0.913	1099.189	0.000	1001.387	1004.965
ar.L1	0.9273	0.047	19.922	0.000	0.836	1.019
ar.L2	-0.1643	0.045	-3.669	0.000	-0.252	-0.077
ma.L1	0.3675	0.044	8.436	0.000	0.282	0.453
sigma2	30.9712	0.982	31.532	0.000	29.046	32.896

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	102.03
Prob(Q):	0.99	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.77	**Skew:**	-0.40
Prob(H) (two-sided):	0.00	**Kurtosis:**	4.02

Fig. 11. Model Summary of ARIMA(2,0,1)

SARIMAX Results

Dep. Variable:	cbl (Mean CBL Pressure-hpa)	**No. Observations:**	1461
Model:	ARIMA(2, 0, 2)	**Log Likelihood**	-4581.809
Date:	Sun, 31 Dec 2023	**AIC**	9175.618
Time:	21:22:15	**BIC**	9207.339
Sample:	01-01-2019	**HQIC**	9187.451
	- 12-31-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	1003.1763	0.919	1091.488	0.000	1001.375	1004.978
ar.L1	0.8759	0.281	3.119	0.002	0.326	1.426
ar.L2	-0.1255	0.216	-0.581	0.561	-0.549	0.298
ma.L1	0.4191	0.281	1.491	0.136	-0.132	0.970
ma.L2	0.0273	0.142	0.193	0.847	-0.251	0.305
sigma2	30.9711	0.988	31.337	0.000	29.034	32.908

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	100.99
Prob(Q):	0.98	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.77	**Skew:**	-0.40
Prob(H) (two-sided):	0.00	**Kurtosis:**	4.01

Fig. 12. Model Summary of ARIMA(2,0,2)

the given dataset, there are no seasonality components so this model is not fit for our dataset.

D. Summary

When Performing a time series in part 1 the author applies a total four models Simple moving average, Exponential Smoothing Model, double Exponential Smoothing model, and

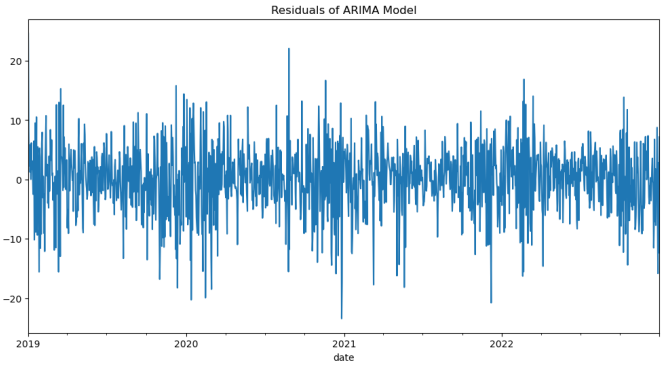

Fig. 13. Residuals Plot of ARIMA Model

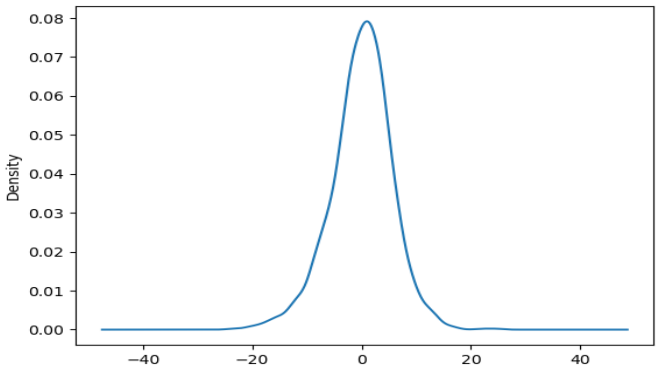

Fig. 14. Residuals Normal Distributed Plot of ARIMA Model

ARIMA. From all the models the ARIMA performed the better with the lowest RMSE value.

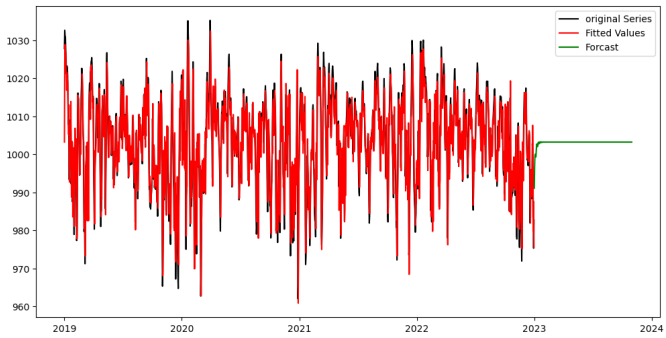

Fig. 15. Forcasting Plot of ARIMA model

II. PART B- LOGISTIC REGRESSION

Dataset 2: Cardiac Data

A. Data Description

Logistic Regression is a model that is used for performing binary classification it is a type of predictive modeling. The binary classification means the output must be a Yes/ No, True/False, or 1/0. The graph of logistic regression is different from the linear regression, in the linear regression graph is

linear but the logistic regression has an S-shaped curve graph. The center point of the S-shaped curve shows the binary classification and converts any numerical value into 0 and 1. The author performs the logistic regression in the cardiac.csv dataset, the dataset has information of the cardiac condition of the participants depending on various factors Age, Weight, etc. The dataset is not too big it has only 100 participants and 5 columns (Age, Weight, Gender, fitness_score, Cardiac condition), the dependent variable is cardiac condition and all other 5 columns are the independent variables. So, the author perform the logistic regression model on this dataset to predict the Cardiac condition.

The Equation of Logistic Regression is :

$$E(Y) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_k X_k}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_k X_k}}$$

B. Descriptive Statistics

Performing the Descriptive Statistics we first import the data using the panda function pd.read_csv. After importing the data, the author checks there is a caseno column in the dataset that shows the count of the data which is not for use for our analysis and descriptive statistics so the author deletes the caseno column. There are two columns (Gender and Cardiac condition) author converts both columns in numerical order using the .replace function. Now the author finds the descriptive of the cardiac data which includes 'count,' which denotes the number of participants in the data, 'mean,' which denotes the participant's average, 'Standard Deviation (std),' which denotes the spread of the values around the average value, 'min,' minimum value represents the lowest values of the cardiac data, 25% (first quartile), 50% (Middle values), 75%(third quartile), 'max,' maximum value represents the highest values of the cardiac data and these all are shown in figure 16.

	count	mean	std	min	25%	50%	75%	max
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
gender	100.0	0.6300	0.485237	0.00	0.0000	1.00	1.0000	1.00
fitness_score	100.0	43.6298	8.571306	27.35	36.5950	42.73	49.2650	62.50
cardiac_condition	100.0	0.3500	0.479372	0.00	0.0000	0.00	1.0000	1.00

Fig. 16. 5-point Summary

After doing the Descriptive analysis of the data, create the correlation between the columns using the .corr() function. In the Figure 17 correlation chart, +1 shows the highest +ve correlation and –0.2 shows the –ve correlation.

Before applying the logistic model to the dataset it is important to check that any outliers are present in the data for the outliers author created the box plot charts which are shown in Figure 18.

In the age column shows few outliers but we did not remove the outliers from the age column because the dataset is small only 100 participants and does not give any effect while creating the model.

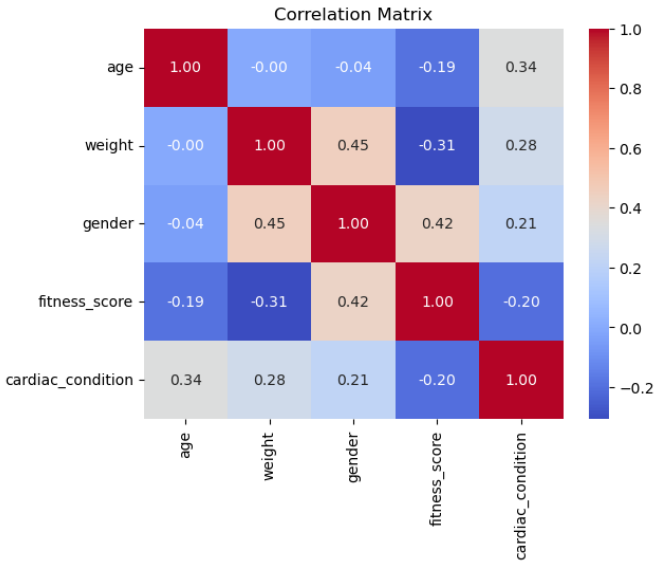

Fig. 17. Correlation Plot

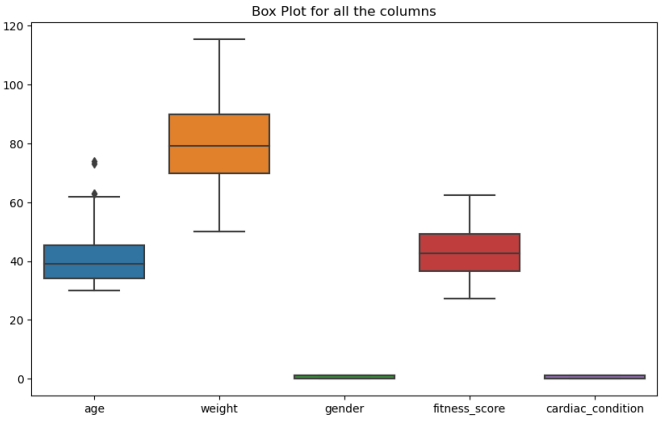

Fig. 18. Outliers

C. Model Process

Model

In creating the model first author divided the data into independent and dependent variables x and y, after the author split the data into train and test data. Secondly performing the feature selection using the standardscalar() function helps the author to improve the performance of the model.

Finally, Fit our first model using sm.Logit class which comes from the Statsmodels, The function sm.add_constant(x_train) which is pass in the sm.Logit class is to account for the intercept in the linear combination of independent variables. Sm.logit class also takes the dependent variable y_train as a parameter and the apply a.fit() method function which calculates the maximum likelihood estimation. To view the summary table of the model author uses the .summary() function which give the DF Residual, Number Observations, Pseudo R-squared, Log-Likelihood, coffeicents, standard error, p-values, etc which all are shown

in figure 19.

Logit Regression Results

Dep. Variable:	y	No. Observations:	70
Model:	Logit	Df Residuals:	65
Method:	MLE	Df Model:	4
Date:	Sat, 30 Dec 2023	Pseudo R-squ.:	0.1886
Time:	21:02:55	Log-Likelihood:	-36.514
converged:	True	LL-Null:	-45.004
Covariance Type:	nonrobust	LLR p-value:	0.001951

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.9549	3.794	-0.252	0.801	-8.392	6.482
x1	0.0686	0.031	2.241	0.025	0.009	0.129
x2	0.0095	0.025	0.382	0.703	-0.039	0.058
x3	1.6267	0.956	1.701	0.089	-0.248	3.501
x4	-0.1006	0.057	-1.752	0.080	-0.213	0.012

Fig. 19. Model Summary

Looking at the p-values of the variables only 1 variable x1 is significant if the author takes the alpha value 0.05, The x1 has only 0.025 that why it was significant. The statistical model does not produce the log odds and the expression fit.params is use to estimate coefficients from the logistic regression model after using the numpy: np.exp(fit.params) applied to exponentiate each of these coefficients. It will give the odds ratio links with each independent variable. The log odds associated with the first coefficient is approx 0.427, it means that one unit increase in the variables is associated with a 0.427 times increase.

Performing the Wald test author examines the significance of individual coefficients associated with predictor variables. The author specifies the null hypothesis whether the coefficients for the variable are equal to 0. Wald test gives the statistic values which is approximately 12.20 and a p-value of approximately 0.015, it is important to determine the statistical significance of the null hypothesis and the last thing is the degree of freedom which is 4. The p-value is less than the significance level of 0.05 author rejects the null hypothesis and concludes that at least one of the four coefficients is greater than 0.

D. Model Performance and Summary

Finally, the author predicts the probabilities of the model using.predict() function and using a list comprehension which converts the predicted probabilities into binary prediction (y_pred). If the y_pred is greater than 0.5 it marks the label 1 otherwise is marked 0. At last, the author creates and displays the confusion matrix which evaluates the performance of the logistic model where the True positive(TP) 17, False positive(FP) 2, False negative(FN) 6, and True negative(TN) are 5 shown in figure 20. after this figure 21 shows,

the summary of the classification report which tells about the accuracy of the model, precision, recall, and f1-score. To find the accuracy we have one function accuracy_score(). The accuracy of this logistic model is 0.73 and the balanced_accuracy_score comes 0.67.

III. Conclusion

In conclusion, this report employed a comprehensive approach to analyze the given datasets (Weather data and Cardiac data), utilizing both time series analysis and logistic regression models. In the time series analysis (Part A), applying simple moving average, exponential models(Simple and Double Exponential), and ARIMA assess their effectiveness in capturing patterns and trends. Among these, the ARIMA model emerged as the most suitable, showcasing superior performance with a smaller Root Mean Square Error (RMSE) compared to the other models. It is important to notice, the absence of seasonality in the dataset led to the exclusion of the SARIMA model. In Part B, logistic regression was employed to conduct a thorough statistical analysis. Various aspects, including the identification of a 5-point summary table, outliers, and correlation assessment, were considered to build a good model. Despite the presence of a limited number of columns in the dataset and the p-values is less then a significant value of 0.05, so decision was made not to apply any dimensionality reduction technique in the logistic regression.

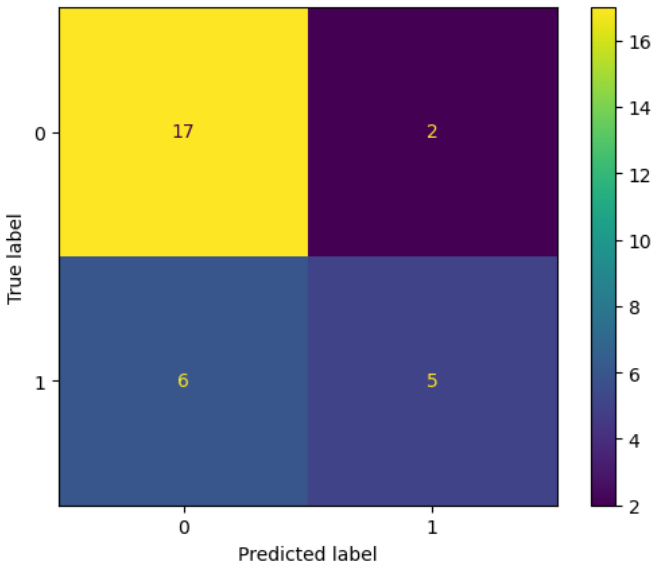

Fig. 20. Confusion Matrix

	precision	recall	f1-score	support
0	0.74	0.89	0.81	19
1	0.71	0.45	0.56	11
accuracy			0.73	30
macro avg	0.73	0.67	0.68	30
weighted avg	0.73	0.73	0.72	30

Fig. 21. Accuracy Metrices