TIME SERIES ANALYSIS AND LOGISTIC REGRESSION FOR STATISTICAL DATA ANALYSIS

Harini Manjunatha (x22169288)
School of Computing,
National College of Ireland, Dublin, Ireland
x22169288@student.ncirl.ie

***Abstract*— This study presents an examination of two statistical modeling techniques time series analysis and logistic regression, both implemented in Python and R programming language respectively. Part A investigates several time series models, such as , SARIMA, exponential smoothing and seasonal models, and shows how to choose the optimum model using AIC and BIC. It also forecasts and assess the model's accuracy with metrics such as MAE, MSE, RMSE, R2, and MAPE. In Part B, using binary logistic regression, we investigate the relationship between a binary response variable and a set of predictor factors also illustrate how to fit the logistic regression model using maximum likelihood estimation and how to evaluate model performance using the confusion matrix.**

PART A : TIME SERIES ANALYSIS.

I. OBJECTIVES

The objective is to perform time series analysis on the diabetic dataset using Python programming language. The analysis includes exploratory data analysis, visualizations, and creating various time series models, such as SARIMA, exponential smoothing, and simple time series models. Each model's performance is evaluated using metrics such as Mean square error, mean absolute error, and root mean square error, and the best model is chosen. The residuals of the chosen model is further evaluated for autocorrelation using the Durbin-Watson statistical tests. Finally, the chosen model is utilized to forecast future time series values.

II. DEFINITION AND TERMINOLOGY

A. Time Series Analysis: Time series analysis is a statistical technique for studying and modeling time-series data. Time series analysis seeks patterns, trends, and correlations in time series data and uses this information to estimate future series values. This technique is commonly used to examine the behavior of time-dependent data and to affect decision-making processes in fields such as economics, finance, engineering, and social sciences.

B. ARIMA/SARIMA:

ARIMA models consist of three parts, differencing autoregression (AR), and moving average (MA). Autoregression is a time series linear regression against previous values. Differencing is a technique for removing any trend or seasonality from time series data.

Moving average includes representing the time series error term as a linear combination of previous error terms.

SARIMA models are ARIMA extensions that include seasonal components. A time series' seasonal component refers to patterns that occur at regular intervals, such as daily, weekly, or monthly. The seasonal component is incorporated into SARIMA models by using additional autoregressive,

differencing, and moving average terms that describe the seasonal patterns in the data.

C. Exponential smoothing: It is a prominent time series analysis approach for forecasting future values based on past data. Exponential smoothing works by weighting recent observations more heavily and less heavily than older observations. The weight assigned to each observation is determined by a smoothing parameter (alpha parameter), which governs how quickly the weight of previous data diminishes over time.

I. DESCRIPTION OF DATA SET

The dataset includes two time series, one for monthly average temperatures in Armagh from January 1844 to December 2004, and the other for yearly average temperatures from 1844 to 2004. The dataset is based on the work of the University of East Anglia's Climate Institute which provides a long-term record of temperature measurements in Armagh that is used to evaluate historical trends and patterns in temperature data.

Monthly

	x
0	4.5
1	2.4
2	4.8
3	9.1
4	10.9
...	...
1927	15.4
1928	13.2
1929	8.6
1930	8.2
1931	6.2

Yearly

	x
0	8.5
1	8.3
2	9.7
3	8.9
4	8.5
...	...
156	9.2
157	8.8
158	9.5
159	9.3
160	9.5

Figure 1: Monthly and yearly timeseries data

The Figure 1 is the datasets of monthly and yearly temperatures in Armagh from 1844 to 2004. Figure 2 represents the description of monthly and yearly time series data which will contain count, mean, standard deviation, quantile, min and max values.

Monthly

	x
count	1932.000000
mean	8.500776
std	3.825564
min	-0.900000
25%	5.300000
50%	8.200000
75%	12.100000
max	17.200000

Yearly

	x
count	161.000000
mean	8.488820
std	0.525475
min	6.700000
25%	8.200000
50%	8.500000
75%	8.800000
max	9.700000

Figure 2: Description of Monthly and yearly timeseries data

III. DATA VALIDATION

The data validation includes checking if any NAs or null value present. Figure 3 shows that there are no null values present in neither monthly nor yearly datasets.

Monthly

	0
x	0

Yearly

	0
x	0

Figure 3 : null check of monthly and yearly dataset

IV. DATA PREPROCESSING

In this section the monthly and yearly frequency range is created for monthly and yearly data from January 1844 to December 2004. The figure shows the resultant dataset after mapping their range.

Monthly

	x
1844-01-01	4.5
1844-02-01	2.4
1844-03-01	4.8
1844-04-01	9.1
1844-05-01	10.9

Yearly

	x
1844-01-01	8.5
1845-01-01	8.3
1846-01-01	9.7
1847-01-01	8.9
1848-01-01	8.5

Figure 4: Preprocessed dataset of Monthly and yearly timeseries.

The data is split into two sets, training and testing. The data up to 2003 is used for training to predict average temperatures for 2004, and the actual data for 2004 is used for validation. Monthly temperature data is used to predict for 12 months, while annual temperature data is used to predict for a year. The ADF (Augmented Dickey-Fuller) test is employed to determine if a time series is stationary. Stationary time series has consistent statistical properties over time, including constant mean and variance, which make it simpler to comprehend and forecast.

The ADF (Augmented Dickey-Fuller) test is used to determine the stationary state of a time series. Because it exhibits constant statistical properties across time, such as a constant mean and variance, a stationary time series is easier to explain and predict. For varying levels of confidence, the ADF test produces test statistics, p-values, and critical values.

If the test statistic at a given confidence level is smaller than the crucial value, we can reject the null hypothesis of non-stationarity and conclude that the time series is stationary.

```
ADF test statistic: -5.00062397412046
p-value: 2.213013539022494e-05
Critical values: {'1%': -3.433787340386774, '5%': -2.8630583903440656, '10%': -2.567578333092222}
```

Figure 5.1: ADF results for monthly time series.

```
ADF test statistic: -1.3523756587488243
p-value: 0.6048831174332471
Critical values: {'1%': -3.473542528196209, '5%': -2.880497674144038, '10%': -2.576878053634677}
```

Figure 5.2: ADF results for yearly time series.

The figure 5.1 and 5.2 represents the ADF test results of monthly and yearly time series respectively. In monthly series data, since test values at all 3 confidence level is less than the critical value, the null hypothesis can be rejected and considered that the time series is stationary. Whereas in the yearly time series, the test value falls within the range of critical value, the null hypothesis cannot be rejected, and time series is not strongly stationary.

V. DATA VISUALISATION

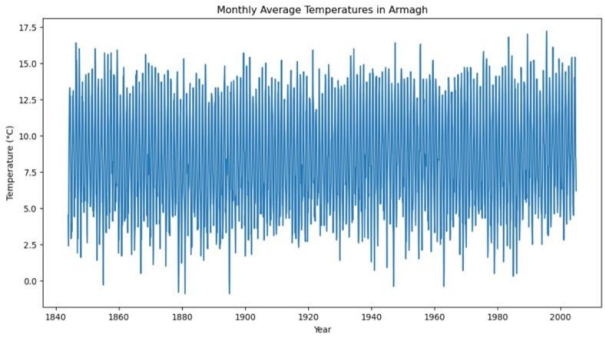

Figure 6.1: Time series plot of monthly temperatures in Armagh.

Figure 6 represents the time series plots of monthly and yearly temperatures at Armagh from 1844 to 2004. The monthly average temperatures in Armagh are depicted in figure 6.1 over time. Each line in the above graph reflects the temperature for each month. This figure provides a more detailed picture of the temperature trend in Armagh, with temperature changes for each month over the years and Figure 6.2 depicts the yearly average temperature in Armagh over time.

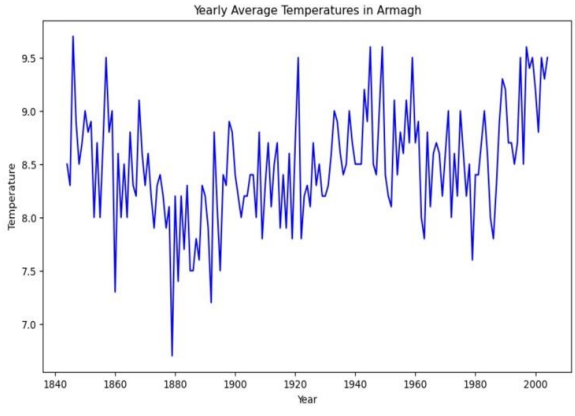

Figure 6.2: Annual temperature time series plot in Armagh.

Figures 7.1 and 7.2 show the seasonal decomposition of additive monthly and yearly time series data. To better comprehend the structure of the series, the decomposition divides the time series into four components: trends, seasonal, original, and residual time series. It also aids in recognizing any recurring patterns or swings that occur within specific months or year, as well as comprehending the residual variation that remains unexplained by trend and seasonality.

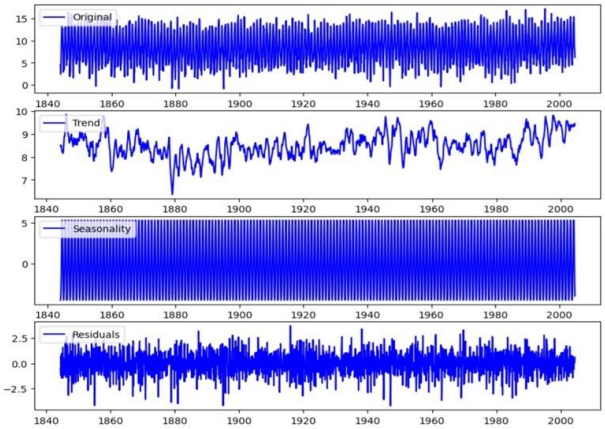

Figure 7.1: Seasonal decomposition of monthly time series

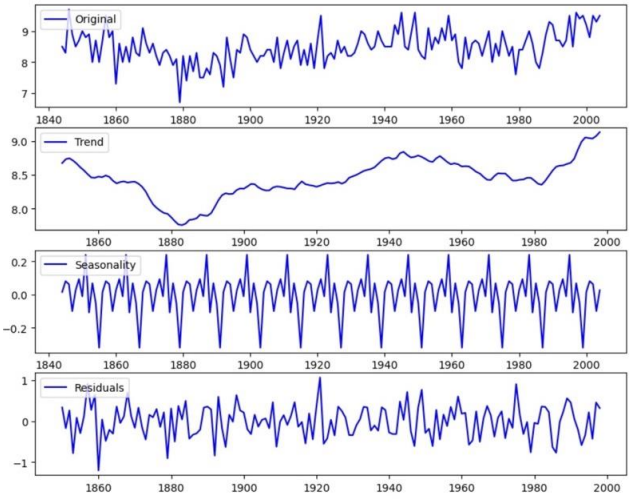

Figure 7.2: Seasonal decomposition of yearly time series

Autocorrelation (ACF) and partial autocorrelation (PACF) plots are useful tools in time series analysis for detecting autocorrelation in data and determining the proper sequence of autoregressive and moving average elements in a time series model. The figure 8.1 and 8.2 auto correlation and partial auto correlation plot for monthly and yearly time series data.

Figure 8.1 depicts seasonal trends with lags corresponding to monthly seasonality. The PACF plot can assist in determining the best order of autoregressive and moving average terms to use when modeling the data.

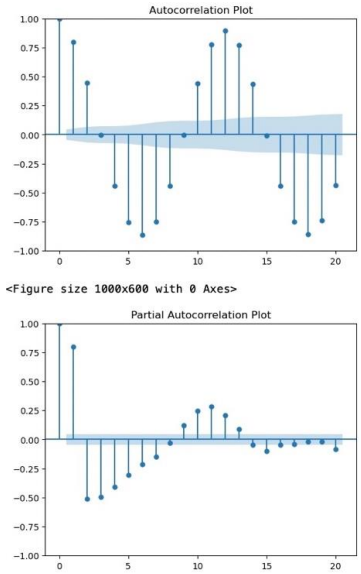

Figure 8.1 ACF and PACF plot of monthly time series.

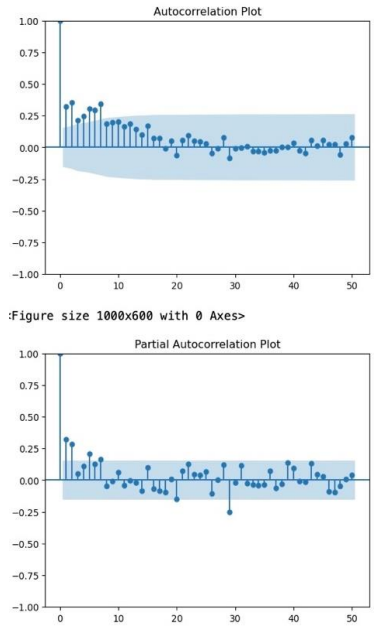

Figure 8.2 ACF and PACF plot of yearly time series.

Figure 8.2 shows that the ACF plot has a substantial association at multiples of the yearly seasonality trend. In addition, there may be long-term dependence that last for several years.

VI.Model Building

1. Exponential smoothing:

Exponential Smoothing is a popular method for time series forecasting that involves giving more weight to recent observations and less weight to older observations. It can handle both additive and multiplicative trends and seasonality. There are several variations of the Exponential Smoothing method, including Holt Linear, Holt Winter, and Damped Trend models.

The Holt Linear model adds a linear trend component to the Exponential Smoothing method, while the Holt Winter model models both trend and seasonality. The Damped Trend model adds a damped trend component to the Exponential Smoothing method, which assumes that the trend will decrease over time. The Exponential Smoothing function from the stats models library can be used to fit these models to the training data and forecast the test data. The results obtained after evaluation of exponential smoothing of monthly and yearly time series data is presented in the figure 9.1 and 9.2 respectively.

2. ARIMA/SARIMA:

ARIMA (Autoregressive Integrated Moving Average) and SARIMA (Seasonal ARIMA) are time series forecasting models that can be used to forecast a variety of time series data. ARIMA models focus on modeling the auto-regressive and moving average components of a time series, while SARIMA models are designed to model the seasonal components of a time series. To fit the SARIMA and ARIMA models in Python, the SARIMAX and ARIMA classes from the stats model's library are utilized. The SARIMA model is applied to monthly data that has shown seasonal patterns, using the order and seasonal order to specify the autoregressive or moving average component in the model, and the seasonal period which is 12 (representing monthly and yearly data). The ARIMA model is applied to yearly data using the order to specify the autoregressive and moving average components. These models are trained on the training data and then used to forecast the test data.

3. Simple time series

Simple time series models are fundamental models used to predict future values in time series data. They are straightforward to comprehend and implement, providing an excellent starting point for modeling time series data. Simple Exponential Smoothing (SES) is one such model that uses a weighted moving average approach to forecast future values. It is suitable for data that does not have any trend or seasonality. The code fits an SES model to the training data and predicts the future values of the test data using the fitted model. Another simple model, Random Walk, assumes that the next value in the time series is the same as the previous value. The code utilizes the seasonal decomposition function to extract the seasonal component from the training data and then shifts the test data by one period to use the previous value as the forecast for the next value. Furthermore, a Seasonal Naive model, which predicts future values based on the previous seasonal value, has been implemented. The code fits the model to the training data and uses it to forecast the future values of the test data.

VII. Model Evaluation

1. The model is evaluated based on 5 different factors mean squared error, mean absolute error, r squared, root mean squared error and MAPE. These are the commonly used evaluation matrix for forecasting models. The figure 9.1 and 9.2 shows the evaluation matrix of monthly and yearly time series data.

	SARIMA	ARIMA	SEASONAL NAIVE	Random walking	Simple Exponential Smoothing
MSE	0.708119	38.625722	0.652387	6.663333	32.450000
MAE	0.703379	5.053288	0.705042	2.183333	4.500000
R2	0.945904	-1.950781	0.950161	0.490960	-1.478992
RMSE	0.841498	6.214960	0.807705	2.581343	5.696490
MAPE	0.076671	0.452053	0.079253	0.271777	0.393774

	Exponential Smoothing	Holt Linear	Holt winter	Damped trend
MSE	0.618375	32.409795	0.624252	32.450000
MAE	0.695483	4.496319	0.700691	4.500000
R2	0.952760	-1.475920	0.952311	-1.478992
RMSE	0.786368	5.692960	0.790096	5.696490
MAPE	0.079997	0.393397	0.080564	0.393774

Figure: 9.1 Evaluation matrix of monthly time series

	SARIMA	ARIMA	SEASONAL NAIVE	Random walking	Simple Exponential Smoothing
MSE	0.033813	0.119338	0.109231	90.25	0.123539
MAE	0.183883	0.345453	0.330501	9.50	0.351481
R2	NaN	NaN	NaN	NaN	NaN
RMSE	0.183883	0.345453	0.330501	9.50	0.351481
MAPE	0.019356	0.036363	0.034790	1.00	0.036998

	Exponential Smoothing	Holt Linear	Holt winter	Damped trend
MSE	0.096422	0.112850	0.121867	0.143347
MAE	0.310519	0.335932	0.349094	0.378612
R2	NaN	NaN	NaN	NaN
RMSE	0.310519	0.335932	0.349094	0.378612
MAPE	0.032686	0.035361	0.036747	0.039854

Figure 9.2: Evaluation matrix of yearly time series

2. *Visualizing* the forecasted value vs actual value to identify the best suited model. Figure 10.1 represents the line graph forecasted values vs actual value of monthly time series data whereas figure 10.2 represents the bar plot of forecasted values vs actual value of yearly time series.

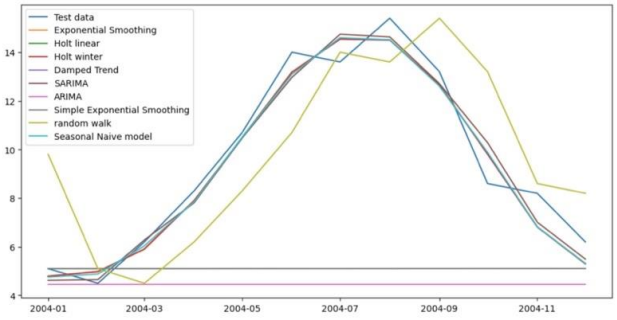

Figure 10.1: Forecast values vs true value for monthly time series

From the figure we can notice that the exponential smoothing model has similar value to actual results and hence we can say that exponential model is the best fit model for monthly time series data.

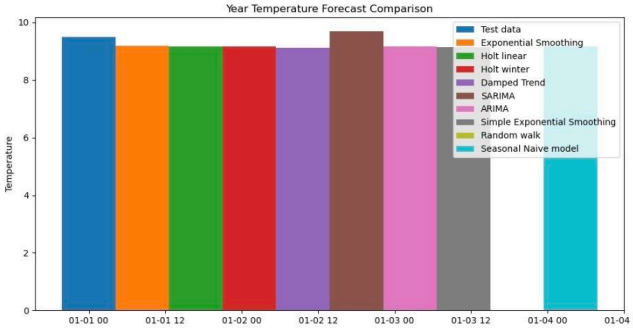

Figure 10.2: Forecast values vs true value for yearly time series

From the above graph we can observe that ARIMA model is the best fitted model for yearly time series analysis .

3. *AIC BIC*:AIC and BIC are statistical methods used to select the best model from a group of candidate models. AIC uses maximum likelihood estimation to measure the model's quality while considering the number of parameters used in the model. BIC is similar to AIC, but it penalizes models with more parameters more heavily, making it more useful when there are fewer observations than parameters. Both AIC and BIC are used to compare the models and select the one that best fits the data. Based on the evaluation results, we can conclude that exponential smoothing, Holt winter, SARIMA, and seasonal naive models perform better for the given datasets. Therefore, applying AIC and BIC to these models would be useful for selecting the best one.

```
exponential smoothing AIC: 758.9611135390905
exponential smoothing BIC: 847.9224009794398
Holt winter AIC: 760.4526109691475
Holt winter BIC: 854.9739788745186
SARIMA AIC: 6231.741022487875
SARIMA BIC: 6259.401463205583
Seasonal Naive AIC: 6209.69311454078
Seasonal Naive BIC: 6226.354547096805
```

Figure 11.1: AIC BIC results for monthly time series

```
exponential smoothing AIC: -229.96598919547773
exponential smoothing BIC: -180.7632081517365
Holt winter AIC: -229.14711060736903
Holt winter BIC: -176.86915574839398
SARIMA AIC: 198.60329068761786
SARIMA BIC: 211.96743485992738
Seasonal Naive AIC: 224.2674488287198
Seasonal Naive BIC: 233.25908565001214
```

Figure 11.2: AIC BIC results for yearly time series

Based on the monthly trend data results, it appears that the SARIMA, ARIMA, and Exponential Smoothing models have the lowest MSE and RMSE, indicating superior accuracy than the other models. Furthermore, the R2 for these two models is high, indicating a solid fit. According to the metrics shown in the table, the Exponential Smoothing model performs best in monthly data and the ARIMA model performs best in yearly data. Furthermore, the model which has low AIC and BIC values, indicates a greater goodness of fit when compared to the other models.

4. Jarque Bera Examination

The Jarque-Bera test is a statistical test used to detect whether data is regularly distributed or not. It's a goodness-of-fit test that analyzes if the data fits the theoretical normal distribution. The test is based on the skewness and kurtosis of the data and can be used to evaluate whether the data is skewed or has heavy tails.

A small p-value shows skewed data, whereas a big p-value indicates regularly distributed data.

```
6.79812383944469e-14
The residuals are not normally distributed.
```

Fig 13.1: Jarque bera test result of monthly time series

```
0.0
The residuals are not normally distributed.
```

Fig 13.2: Jarque bera test result of yearly time series

The Jarque-bera test yields modest p-values, such as 6.798e-127 and 0, for both monthly and yearly trend data, indicating that there is strong evidence to reject the null hypothesis of normality.

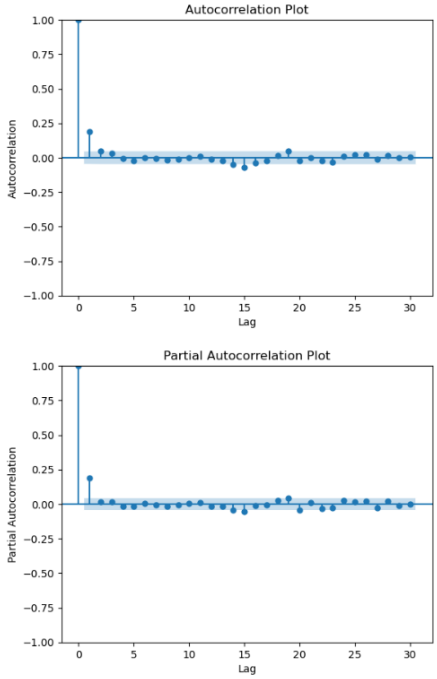

Fig 12.1: Auto-correlation and partial auto-correlation plot of SARIMA model on monthly trend data

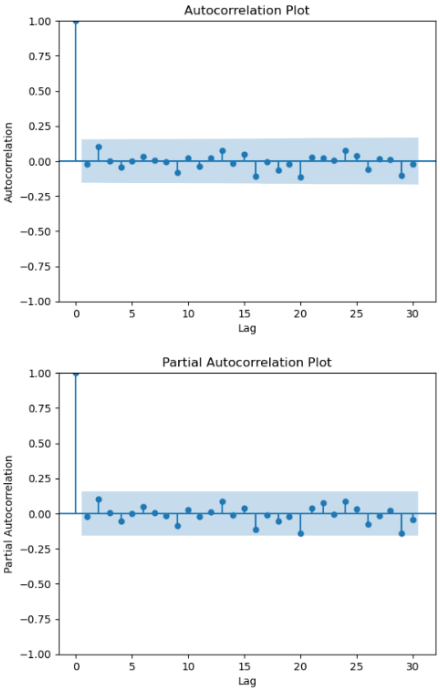

Fig 12.2: Auto-correlation and partial auto-correlation plot of SARIMA model on yearly trend data

The Autocorrelation and partial autocorrelation graphs show that over 90% of the data are inside the upper and lower bands for both monthly and yearly data.

5. Durbin Watson Test

The Durbin-Watson test is a statistical method used to detect the presence of autocorrelation in regression residuals. The Durbin-Watson test can assess whether or not there is a correlation between the residuals of a regression model, which is referred to as autocorrelation. This correlation violates one of linear regression's assumptions, which is that the residuals are independent of one another.

The Durbin-Watson test produces a test statistic that assesses residual autocorrelation.

It is especially effective for detecting first-order autocorrelation, which arises when residuals are correlated with the values immediately preceding them.

```
Durbin-Watson statistic: 0.04
Autocorrelation is present
```

Fig 14.1: Durbin Watson test result of monthly time series

```
Durbin-Watson statistic: 0.00
Autocorrelation is present
```

Fig 14.2: Durbin Watson test result of yearly time series

The Durbin-Watson test result in the monthly trend data is 0.4, indicating high positive autocorrelation in the residuals. This suggests that the regression model's residuals are positively correlated, indicating a pattern or relationship between sequential residuals.

The Durbin-Watson test result for yearly data, on the other hand, is 0.0, indicating complete positive autocorrelation in the residuals. This suggests that the residuals are perfectly correlated, and there is a clear linear relationship between successive residuals.

VIII. Forecast

```
Forecasts for the next year:
 161    9.393073
dtype: float64
```

Fig 15.1: Yearly data ARIMA model forecast of the year 2005.

```
Forecasts for the next 12 months:
 2005-01-01     4.871098
2005-02-01     5.058672
2005-03-01     5.989730
2005-04-01     7.984130
2005-05-01    10.574198
2005-06-01    13.267365
2005-07-01    14.616438
2005-08-01    14.589724
2005-09-01    12.742523
2005-10-01     9.885378
2005-11-01     6.905874
2005-12-01     5.388489
Freq: MS, dtype: float64
```

Fig 15.2: Monthly data Exponential Smoothing model forecast of the year 2005.

The figure 15 represents the future forecast of test results of yearly and monthly time series of their best suited models. ARIMA is the best model for yearly data series, while exponential smoothing is the best model for monthly data series.

VIII. Summary

The dataset was comprehended and its structure was examined. Duplicate data points were removed and necessary preprocessing was carried out to ensure the accuracy of the stationary data. Multiple time series models were constructed and visualized. The best models were identified for each dataset and relevant parameters were evaluated. A series of experiments were conducted to test for normality and correlation in the data.

PART 2: LOGISTIC REGRESSION

I. OBJECTIVES

The objective of this task is to build a binary logistic regression model that can predict diabetes based on blood results, using the "Diabetes Dataset.csv" file. This document offers information such as blood glucose levels, age and BMI of diabetic patients of Iraqi university hospital in the year 2020.

II. DEFINITION AND TERMINOLOGY

LOGISTIC REGRESSION : Logistic regression (LR) is a statistical approach for examining the relationship between a target variable and a number of independent variables. It is commonly used to describe binary outcomes, where the dependent variable might have one of two possible values (e.g., Yes or NO). Logistic regression creates a probability estimate based on the independent variables value.

III. Dataset Description

The dataset contains information on diabetic patients blood samples gathered in 2020 at an Iraqi university hospital. The data contains 14 columns and 1000 rows including age, gender, various blood test results and binary variable indicating weather or not the patient is diabetic. The target variable is termed as class and it has 2 classes YES and NO which indicate weather or not a patient is diabetic

Target variable – CLASS
No of variable – 14
No of rows – 1000

```
> str(df)
'data.frame':   1000 obs. of  14 variables:
 $ ID      : int  502 420 680 634 721 759 636 788 82 132 ...
 $ No_Pation: int  17975 47975 87656 34224 34225 34230 34231 34232 46815 34234 ...
 $ Gender  : chr  "F" "F" "F" "F" ...
 $ AGE     : int  50 50 50 45 50 32 31 33 30 45 ...
 $ Urea    : num  4.7 4.7 4.7 2.3 2 3.6 4.4 3.3 3 4.6 ...
 $ Cr      : int  46 46 46 24 50 28 55 53 42 54 ...
 $ HbA1c   : num  4.9 4.9 4.9 4 4 4 4.2 4 4.1 5.1 ...
 $ Chol    : num  4.2 4.2 4.2 2.9 3.6 3.8 3.6 4 4.9 4.2 ...
 $ TG      : num  0.9 0.9 0.9 1 1.3 2 0.7 1.1 1.3 1.7 ...
 $ HDL     : num  2.4 2.4 2.4 1 0.9 2.4 1.7 0.9 1.2 1.2 ...
 $ LDL     : num  1.4 1.4 1.4 1.5 2.1 3.8 1.6 2.7 3.2 2.2 ...
 $ VLDL    : num  0.5 0.5 0.5 0.4 0.6 1 0.3 1 0.5 0.8 ...
 $ BMI     : num  24 24 24 21 24 24 23 21 22 23 ...
 $ CLASS   : chr  "N" "N" "N" "N" ...
```

Fig. 16. Information of dataset

The above figure is the structure of the data set. The variables of dataset and their datatypes are shown.

```
> summary(df)
       ID          No_Pation          Gender               AGE             Urea             Cr        
 Min.   :  1.0   Min.   :    123   Length:1000        Min.   :20.00   Min.   : 0.500   Min.   :  6.00  
 1st Qu.:125.8   1st Qu.:  24064   Class :character   1st Qu.:51.00   1st Qu.: 3.700   1st Qu.: 48.00  
 Median :300.5   Median :  34396   Mode  :character   Median :55.00   Median : 4.600   Median : 60.00  
 Mean   :340.5   Mean   : 270551                      Mean   :53.53   Mean   : 5.125   Mean   : 68.94  
 3rd Qu.:550.2   3rd Qu.:  45384                      3rd Qu.:59.00   3rd Qu.: 5.700   3rd Qu.: 73.00  
 Max.   :800.0   Max.   :75435657                     Max.   :79.00   Max.   :38.900   Max.   :800.00  
     HbA1c             Chol             TG              HDL              LDL              VLDL       
 Min.   : 0.900   Min.   : 0.000   Min.   : 0.30   Min.   :0.200   Min.   :0.30   Min.   : 0.100  
 1st Qu.: 6.500   1st Qu.: 4.000   1st Qu.: 1.50   1st Qu.:0.900   1st Qu.:1.80   1st Qu.: 0.700  
 Median : 8.000   Median : 4.800   Median : 2.00   Median :1.100   Median :2.50   Median : 0.900  
 Mean   : 8.281   Mean   : 4.863   Mean   : 2.35   Mean   :1.205   Mean   :2.61   Mean   : 1.855  
 3rd Qu.:10.200   3rd Qu.: 5.600   3rd Qu.: 2.90   3rd Qu.:1.300   3rd Qu.:3.30   3rd Qu.: 1.500  
 Max.   :16.000   Max.   :10.300   Max.   :13.80   Max.   :9.900   Max.   :9.90   Max.   :35.000  
      BMI           CLASS          
 Min.   :19.00   Length:1000        
 1st Qu.:26.00   Class :character   
 Median :30.00   Mode  :character   
 Mean   :29.58                      
 3rd Qu.:33.00                      
 Max.   :47.75                      
```

Fig. 17. Description of dataset

Figure 17 shows a description of the dataset, It includes the information on count, mean, standard deviation, quantile, minimum and maximum values for each variable in the dataset.

I. DATA VALIDATION

```
> head(na_values, 25)
       colSums.is.na.df_model..
Gender                        0
AGE                           0
Urea                          0
Cr                            0
HbA1c                         0
Chol                          0
TG                            0
HDL                           0
LDL                           0
VLDL                          0
BMI                           0
CLASS                         0
>
```

Fig 18: Null value check

During data validation, the dataset is checked for null values. Figure 18 shows that our dataset does bot contain any null values or NAs.
The variable ID is removed from the list of independent variables because it makes no difference in diabetes outcomes.

VI. Data Visualization

We interpret data from multiple sources into physical images to obtain detailed information about the record. Data visualization graphically summarizes the statistical overview of the given data. One form of plot in which we can visualize data for greater understanding is the correlation plot. The correlations between the dependent variable and each associated independent variable can be examined. We are plotting the correlation plot for all the variables in the dataset from which we can then explore the relationship between the dependent and independent variables.

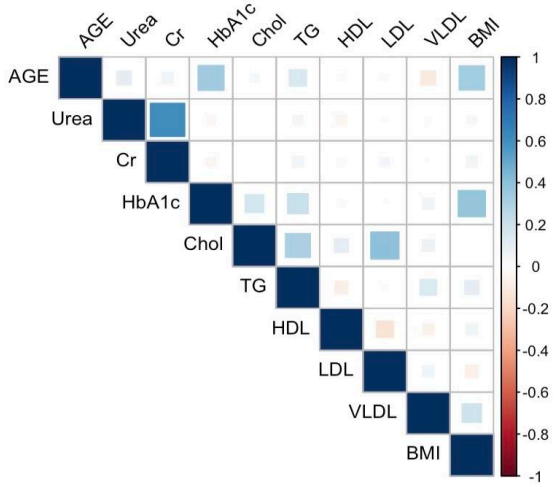

Figure 19: Correlation plot of dataset

The correlation plot of the given dataset is depicted in Figure 19. According to the graph, lighter color reflects the positive correlation whereas the darker color reflects the 0 correlation. The plot shows that No_pation, HDL, and LDL have no association with the target variable.

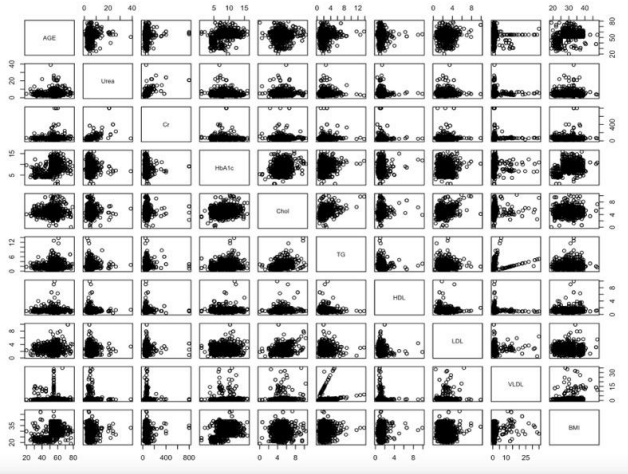

Figure 20: Box plot of dataset

Boxplot visualizes statistical data. It is used to identify outlier points that do not fall inside the data's interquartile range. Figure 20 shows a boxplot depiction of the diabetic dataset.

VII. MODEL BUILDING

This section of the report will outline the construction steps of the logistic regression model. Figure shows the generalized code used to construct Logistic Regression model. The dataset is split into train data and test data, 70% of data is used for training the model and 30 % for testing the result and finding the accuracy of the model. Three different logistic models are built. The best fit model is chosen by comparing their results.

Model1: In Model 1, the logistic regression model is built by considering all the available independent variables to predict the CLASS variable in the dataset. Figure shows the summary of model1.

```
> summary(model_1)

Call:
glm(formula = CLASS ~ ., family = binomial, data = train1)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.17764   0.00000   0.00004   0.00111   2.38636

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -58.46913   12.89775  -4.533 5.81e-06 ***
Gender        0.91596    0.82717   1.107 0.268150
AGE          -0.02739    0.04632  -0.591 0.554256
Urea         -0.07331    0.17433  -0.421 0.674087
Cr            0.01815    0.01976   0.919 0.358233
HbA1c         1.97972    0.49931   3.965 7.34e-05 ***
Chol          0.95442    0.31582   3.022 0.002511 **
TG            2.46336    0.73033   3.373 0.000744 ***
HDL           0.96918    0.63600   1.524 0.127545
LDL           0.83390    0.46489   1.794 0.072852 .
VLDL         -0.07330    0.32053  -0.229 0.819111
BMI           1.52252    0.35515   4.287 1.81e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 425.319  on 662  degrees of freedom
Residual deviance:  51.908  on 651  degrees of freedom
AIC: 75.908

Number of Fisher Scoring iterations: 11
```

Fig. 21.1 Summary of the model 1.

```
> modelOutput(test1,pred_class)
     Metric Value
1  Accuracy 0.954
2 Precision 0.972
3    Recall 0.976
4  F1 Score 0.974
```

Fig. 21.2 Results of the model 1.

The figure 21.1 is the summary of model 1 and 21.2 is the results obtained on evaluastion of model 1. The model's accuracy is 95%, and the F1 score is 0.97, indicating that model 1 is both a good model and the best fit. The model can be further improved by considering only the most significant variables.

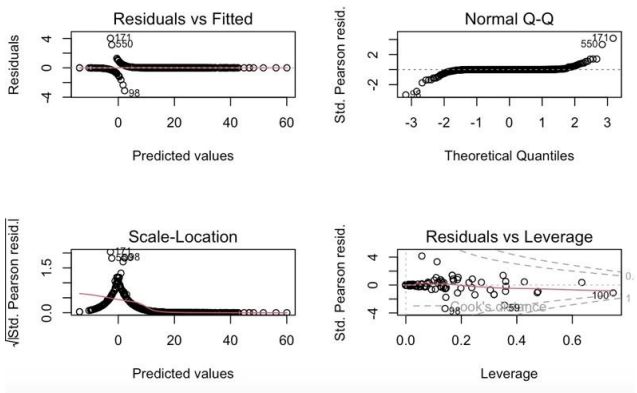

Fig. 21.3 Plot of the model 1.

Model2: The second model is built by considering only significant variables which are determined by the summary of model 1. HbA1c,Chol,TG,BMI are considered as independent variable and CLASS variable is considered as a dependent variable. The summary of the resulting model is shown below.

```
> summary(model_2)

Call:
glm(formula = CLASS ~ HbA1c + Chol + TG + BMI, family = binomial,
    data = train1)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.85934   0.00000   0.00016   0.00198   2.45513

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -46.5876     9.4027  -4.955 7.24e-07 ***
HbA1c         1.7600     0.3959   4.446 8.76e-06 ***
Chol          0.8847     0.2675   3.308 0.000941 ***
TG            2.0073     0.5669   3.541 0.000399 ***
BMI           1.2678     0.2902   4.369 1.25e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 425.319  on 662  degrees of freedom
Residual deviance:  59.066  on 658  degrees of freedom
AIC: 69.066

Number of Fisher Scoring iterations: 11
```

Fig. 22.1 Summary of the model 2.

```
> modelOutput(test1,pred_class)
     Metric Value
1  Accuracy 0.965
2 Precision 0.980
3    Recall 0.980
4  F1 Score 0.980
```

Fig. 22.2 Results of the model 2.

The figure 22.1 is the summary of model 2 and 22.2 is the results obtained on evaluastion of model 2. The accuracy of the model is 96.5%, which has slight improvement than the model 1 with F1, precission and recall value of 0.98. The model 2 can be considered as a best fit model than model 1. The model can still be improved by applying transformations which is implemented in model 3.

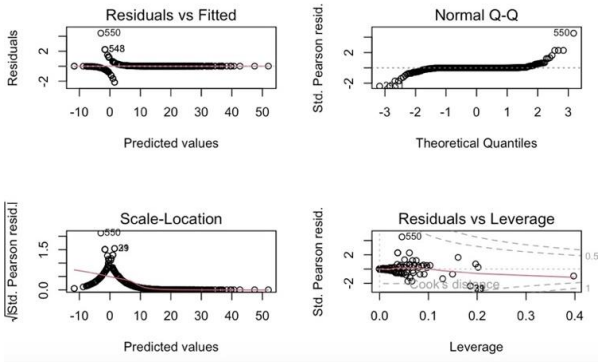

Fig. 22.3 Plot of the model 2.

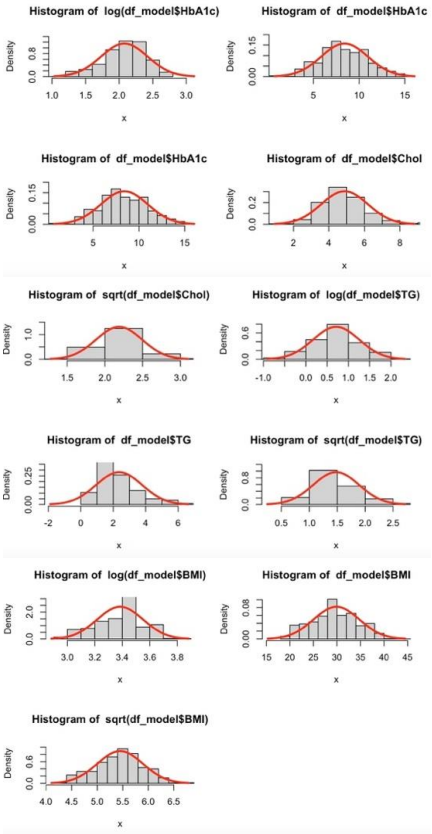

Figure 24 : Histogram plot of significant variable

The figure 24 represents the histogram plot of significant independent variable. By considering the above figure, the log sqrt transform is applied to the third model to further improve the accuracy.

```
> summary(model_3)

Call:
glm(formula = CLASS ~ log(HbA1c) + log(Chol) + log(TG) + log(BMI),
    family = binomial, data = train1)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.00427   0.00009   0.00072   0.00525   2.64271

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -117.891     23.748  -4.964 6.90e-07 ***
log(HbA1c)     8.801      1.843   4.776 1.79e-06 ***
log(Chol)      3.336      1.155   2.889  0.00386 **
log(TG)        3.615      1.053   3.432  0.00060 ***
log(BMI)      30.739      6.489   4.737 2.17e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 425.319  on 662  degrees of freedom
Residual deviance:  66.361  on 658  degrees of freedom
AIC: 76.361

Number of Fisher Scoring iterations: 10
```

Figure: 25.1 The summary of the model 3

The summary of model 3 is shown in the figure4 which has accuracy of 95% which is lesser than the accuracy of the previous model. Hence we can consider that model2 is the best fit model.

```
> modelOutput(test1,pred_class)
     Metric Value
1  Accuracy 0.954
2 Precision 0.972
3    Recall 0.976
4  F1 Score 0.974
```

Figure: 25.2 Results of model 3

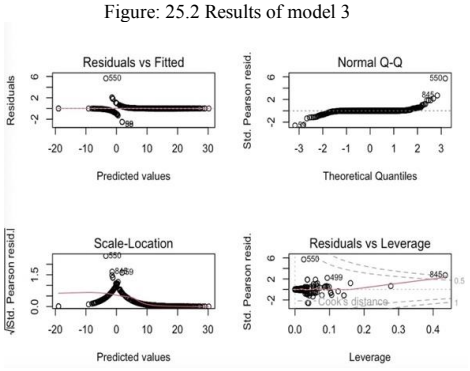

Figure: 25.3 Results of model 3

Parameter to check goodness of fit

Hosmer and lemeshow test:

The Hamster Lemeshow test determines whether the observed frequencies in each group deviate considerably from the expected frequencies predicted by the model. If the difference is considerable, i It shows that the model does not fit the data well.

If p>0.05, the model is deemed to be fit.

```
> hl_test <- hoslem.test(train1$CLASS, predict(model_2, type = "response"))
> hl_test

        Hosmer and Lemeshow goodness of fit (GOF) test

data:  train1$CLASS, predict(model_2, type = "response")
X-squared = 0.33821, df = 8, p-value = 1
```

Figure 26: Hosmer lemeshow test results

The P value in the figure is greater than 0.05, implying that the model is the best model.

VIII.PREDICTION.

The probability of class P getting diagnosed as diabetic from the best suited model is 73% which is displayed in the below figure.

```
> cat("Probability of diagnosing the 'P' class cases as diabetic: ", prob_diabetic, "\n")
Probability of diagnosing the 'P' class cases as diabetic:  0.7396979
```

Figure 27: Probability of test results

IX.SUMMARY.

The three distinct logistic models were created and analyzed as part of this study to find the best fit model. Based on our findings, model 2 is the best-fitting logistic regression model for predicting the presence of diabetes in patients based on blood test data. On the test dataset, the model has an accuracy of 96%, indicating that it is a solid predictor of diabetes. Using the developed model, the likelihood of

patients developing diabetes was estimated to be 73%. This model can assist healthcare personnel in identifying patients at risk of developing diabetes and providing early intervention to prevent or treat the disease.

X.REFERENCES.

[1]. A. Yamini and K. S. Rekha, "Improved Accuracy for Identifying At-Risk Students at Different Percentage of Course Length using Logistic Regression Compared with K-Nearest Neighbour Model," *2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF)*, Chennai, India, 2023.

[2]. Fleiss, J. (1981). Statistical Methods for Rates and Proportions (Second Edition). New York: Wiley.

[3]. T. C. Lwin, T. T. Zin and P. Tin, "Predicting Calving Time of Dairy Cows by Exponential Smoothing Models," *2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)*, Kobe, Japan, 2020.

.