Statistical Analysis for Time Series and Binary Logistic Regression

Arnab Hati
Master of Science in Data Analytics
National College of Ireland
Dublin, Ireland
x22107321@student.ncirl.ie

Abstract* - This study demonstrates the use of time series analysis and logistic regression using R and Python, respectively. The report explains the analytical process from raw data processing, transformation, scaling, model selection, and the correctness of the final model. The first section of the study focuses on time series analysis, where two datasets were analyzed using several models. The "Armagh temperature" dataset has the best accuracy in the study, which includes forecasting and model selection. The Logistic Regression study, which involves examining a dichotomous dependent variable in the "Diabetes" dataset, is highlighted in the second half of the paper. An assumption check and explanation were done once the model was altered to reach maximum accuracy. The Time Series Analysis section used various models to analyze two datasets and determine their accuracy. The models were selected based on their appropriateness for each dataset and their ability to forecast future values. The "Diabetes" dataset, where the model was altered to increase accuracy, was the subject of the Logistic Regression analysis part. An explanation of the analysis was given, and the assumptions were verified. The study wraps up by summarizing the investigation's findings and methods, emphasizing the accuracy attained in both time series analysis and logistic regression.

Keywords* - Time Series Analysis, Logistic Regression, R, Python, Transformation, Scaling, Assumptions, Model Selection, Accuracy, Forecasting.

I. TIME SERIES ANALYSIS

A. Introduction

A strong statistical technique for analyzing and predicting time-dependent data is time series analysis. It entails looking through a dataset gathered over time to find cycles, trends, and patterns [1]. Numerous disciplines, including economics, finance, weather prediction, and other sciences, frequently employ time series analysis. Two datasets that depict the typical temperatures in Armagh over time will be subjected to time series analysis in this research.

The aim of this initiative is to estimate and present appropriate models for both monthly and yearly time series. Using suitable visualizations, we will first undertake a preliminary analysis of the nature and elements of the raw time series. Exponential smoothing, ARIMA/SARIMA, and simple time series models are the three categories from which we shall estimate and analyze acceptable time series models. To assess the accuracy and efficacy of the models, appropriate diagnostic tests and checks will be carried out. [2]

In order to predict the average temperatures for 2004, we will utilize the data up to and including 2003 as a training set. The yearly temperature data will be used to predict for one year, and the monthly temperature data will be used to forecast for 12 months. The projections will be assessed in comparison to the actual data for 2004. We will then decide on the best model for the series and offer input on how well it works as a forecasting tool [3]. This project's overall goal is to show how effective time series analysis is at analyzing and predicting time-dependent data, particularly in the context of climate research.

B. Dataset

Two data files have been produced and are posted on Moodle based on the research of the Climate Institute of the University of East Anglia: A monthly time series of the average temperatures in Armagh from January 1844 to December 2004 is shown in the datafile "nitm18442004.csv". A reduced version of the same data, "nity18442004.csv," provides a time series of annual average temperatures from 1844 to 2004. There are differences between the two time series.

- Dataset 1: "nitm18442004.csv"
- Dataset 2: "nity18442004.csv"

C. Software Used

- Dataset 1: RStudio 1.3.1093 version is used.
- Dataset 2: Jupter Notebook and Python.

D. Objective

to determine which of the three models—Exponential Smoothing, ARIMA/SARIMA, and Simple Time Series Models—is the best suitable. a preliminary analysis of the nature and elements of the raw time series, making use of relevant visualizations. As a training set, use the data up to and including 2003 to predict the average temperatures for 2004, then use the actual data for 2004 as a test set to assess the accuracy of the forecast for the year. To predict for 12 months, use the monthly temperature data, and to forecast for a year, use the annual temperature data. Compare the 2004 predictions to the actual facts.

E. Analysis and Visualisation

I. Dataset 1

- A time series plot is plotted from the raw data as shown in Fig.1. which shows the pattern of the temperature every twelve months from 1844 to 2004.

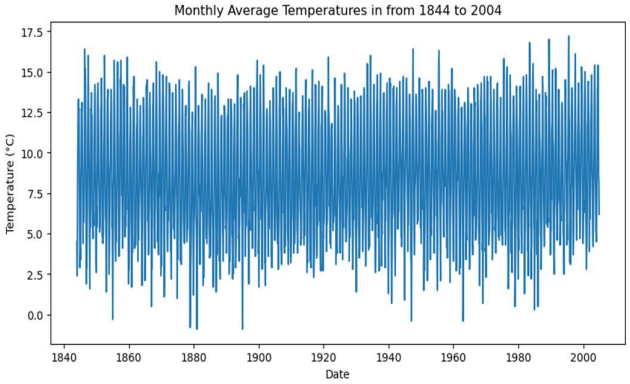

Fig. 1

- Two statistical methods that are frequently used in time series analysis to find stationarity in a time series are the rolling statistic and the Dickey-Fuller test. Rolling statistics are used to compare the mean and variance of a time series over time. If the rolling statistics remain constant, it indicates that the time series is stationary, while if they change, it indicates that it is non-stationary as plotted in Fig. 2. The Dickey-Fuller test is a statistical hypothesis test used to determine whether a time series is stationary or non-stationary. It involves estimating the parameters of an autoregressive model and testing whether the residuals are stationary. If the residuals are stationary, the null hypothesis is rejected, indicating that the time series is stationary shown in Fig 3.

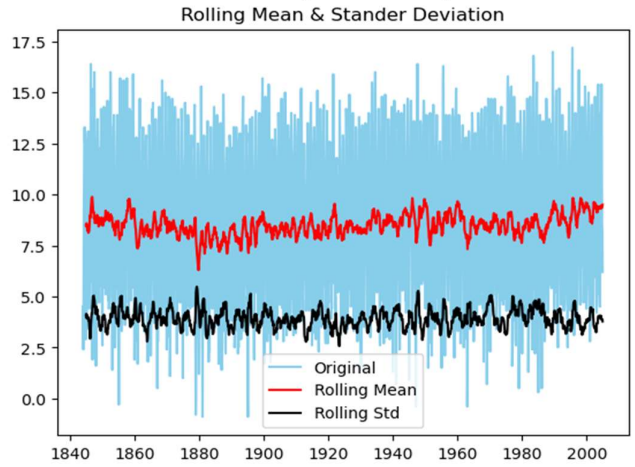

Fig. 2

```
Result of Dickey-Fuller Test:
Test Statistic                  -5.000624
P-value                          0.000022
#Lags Used                      26.000000
Number of Observations Used   1905.000000
Critical Value (1%)             -3.433787
Critical Value (5%)             -2.863058
Critical Value (10%)            -2.567578
```

Fig. 3

- Seasonality in time series refers to the existence of recurring patterns or variations that do so throughout the course of time. In a time, series plot, seasonality may be seen by looking for repeated patterns at regular intervals, as seen in Fig 4. Seasonality in the time series is present if the pattern's amplitude and period remain stable throughout time.

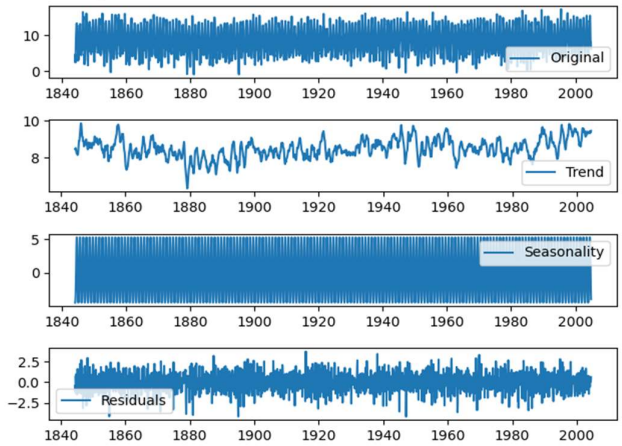

Fig. 4

- Exponential smoothing is a popular time series forecasting method used to make predictions based on past observations. It works by giving more weight to recent observations and less weight to older observations, using an exponentially decreasing function [4]. The error or residual is multiplied by a smoothing factor (alpha) and added to the forecast to generate a new forecast for the next time period.
- I implemented Holt-Winters' Seasonal Exponential Smoothing (HW-SES), a prominent time series forecasting technique, on this dataset to smooth out time series data that contain a trend but no seasonal component. Using two smoothing variables (alpha and beta) to regulate the weights assigned to the level and trend components, it extends the capabilities of the straightforward exponential smoothing approach to include a trend component. described in Figure 5-7.

```
Holt-Winters' Seasonal Exponential Smoothing RMSE: 0.786
Holt-Winters' Seasonal Exponential Smoothing MAE: 0.695
Holt-Winters' Simple Exponential Smoothing R2: 0.953
```

Fig 5

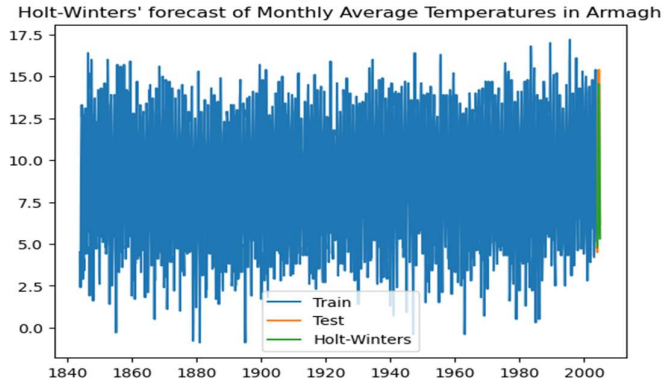

Fig 6

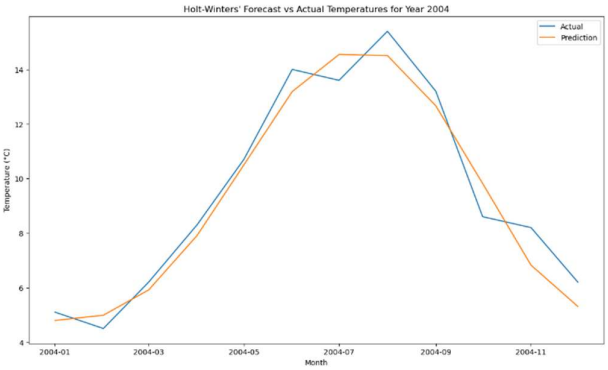

Fig 7

- The seasonal naive model, one of the Simple time series models, was the model under consideration for further investigation. The naive model and the mean model were evaluated as well shown Fig 8-10.
- The seasonal naive model is a simple forecasting method used to predict the value of time series data with seasonal patterns. It assumes that the seasonal pattern remains constant over time and is often used as a baseline model for comparison with more complex forecasting methods.

```
Seasonal Naive Model:
Seasonal Naive Model RMSE: 0.73
Seasonal Naive Model R2 Score: 0.96
```

Fig 8

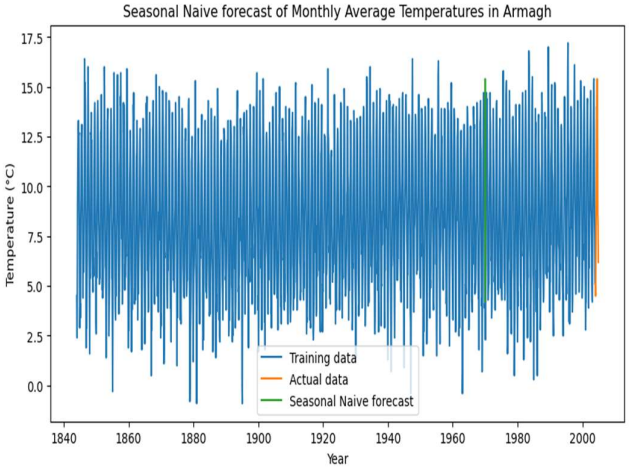

Fig 9.

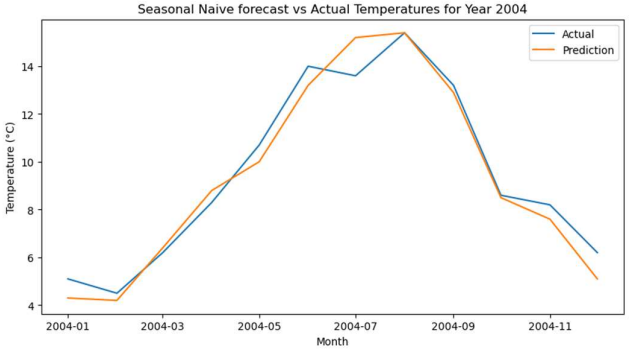

Fig 10

- The autocorrelation structure of time series data may be understood using an autocorrelation plot, sometimes referred to as an ACF plot, like the one in Figure 11. The term "autocorrelation" describes the relationship between a time series and a delayed version of it.

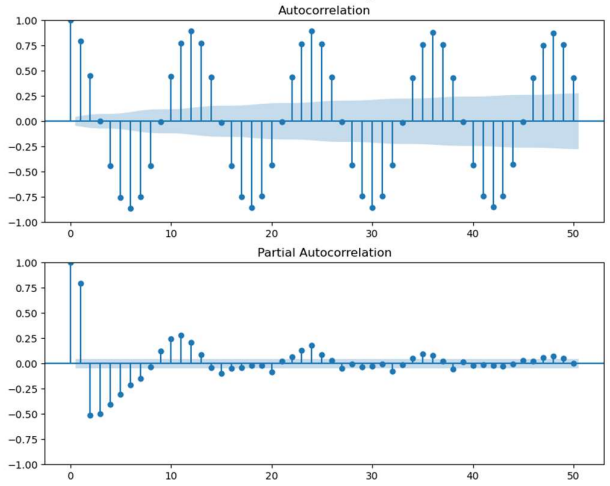

Fig 11.

- The popular time series forecasting model SARIMA (Seasonal Autoregressive Integrated Moving Average) extends the ARIMA model to take into account seasonal patterns in the data.
- The four main parameters that make up the SARIMA model are p, d, q, and P, D, Q, s. The order of the autoregressive, integrated, and moving average components is represented by the same three parameters—p, d, and q—as in the ARIMA model. The parameters P, D, and Q reflect the order of the seasonal autoregressive, seasonal integrated, and seasonal moving average components, respectively. The parameter s specifies the duration of the seasonal cycle in the data (e.g., if the data is monthly, s=12 for yearly seasonality). The next model in our analysis is the SARIMA model. First, we implemented the auto SARIMA and then check the model summary. To assess the model's performance, we used various visualization techniques as residuals plot, Q-Q plot and values for the R2 score, RMSE, etc shown in Fig. 12- 16

SARIMAX Results

Dep. Variable:	y	No. Observations:	1920
Model:	SARIMAX(3, 1, 1)x(2, 0, 1, 12)	Log Likelihood	-3197.157
Date:	Thu, 11 May 2023	AIC	6412.315
Time:	14:14:18	BIC	6462.351
Sample:	01-01-1844 - 12-01-2003	HQIC	6430.725
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	5.453e-06	6.31e-05	0.086	0.931	-0.000	0.000
ar.L1	0.2692	0.024	11.207	0.000	0.222	0.316
ar.L2	0.0442	0.024	1.808	0.071	-0.004	0.092
ar.L3	0.0090	0.025	0.353	0.724	-0.041	0.059
ma.L1	-0.9797	0.009	-114.874	0.000	-0.996	-0.963
ar.S.L12	0.8707	0.028	31.651	0.000	0.817	0.925
ar.S.L24	0.1160	0.026	4.392	0.000	0.064	0.168
ma.S.L12	-0.8561	0.018	-47.175	0.000	-0.892	-0.821
sigma2	1.7135	0.057	30.075	0.000	1.602	1.825

Ljung-Box (L1) (Q):	0.06	Jarque-Bera (JB):	98.84
Prob(Q):	0.80	Prob(JB):	0.00
Heteroskedasticity (H):	0.81	Skew:	-0.35
Prob(H) (two-sided):	0.01	Kurtosis:	3.87

Fig 12 Model Summary

```
SARIMA RMSE: 0.847
SARIMA MAE: 0.755
SARIMA R2: 0.945
```

Fig 13. Results

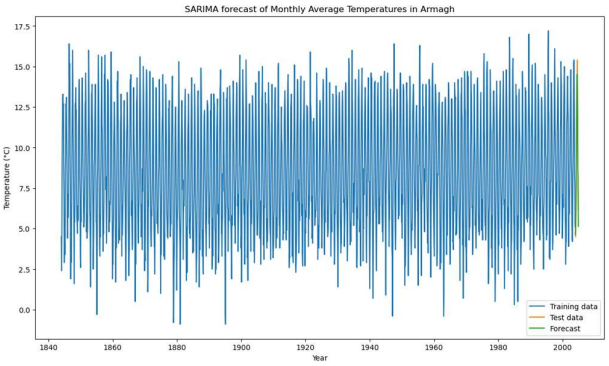

Fig 14.

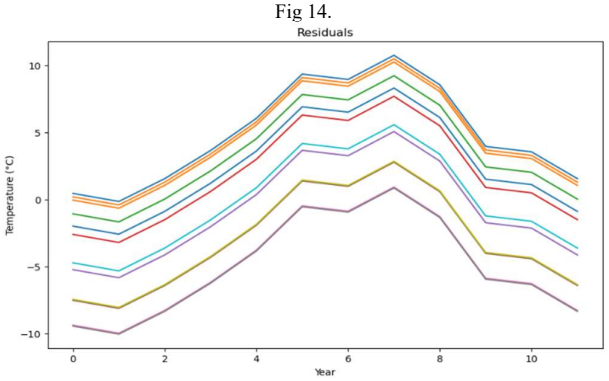

Fig 15. Residuals Plot

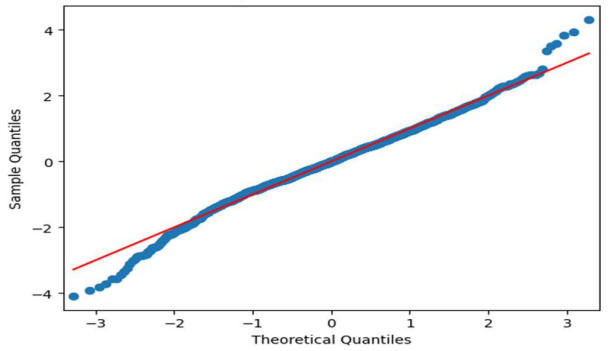

Fig 16. Q-Q Plot

II. Dataset 2

- The dataset is titled nity18442004.csv, and R was used for the analysis. For these two libraries, "tseries" and "forecast" were imported.
- In order to determine the nature of the raw data, a time series plot and a histogram were created, as illustrated in Figures 16 and 17.

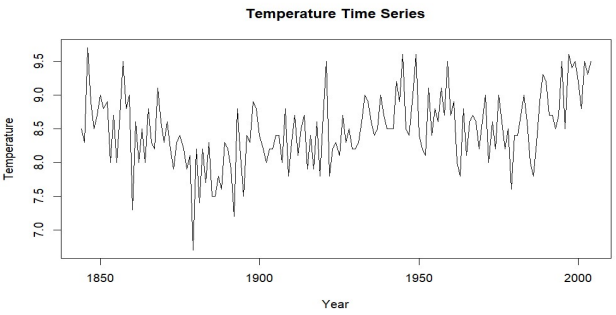

Fig17

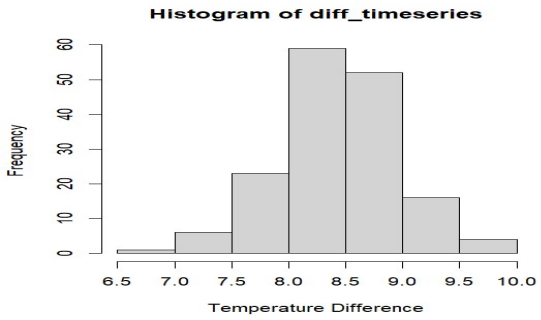

Fig. 18

- The ADF test is a statistical test that can be used to determine if a time series is stationary or not. If the p-value is less than the chosen level of significance (e.g., 0.05), then you can reject the null hypothesis that the time series is non-stationary and conclude that it is stationary. Otherwise, you cannot reject the null hypothesis and the time series is likely non-stationary as in Figure 19.

```
        Augmented Dickey-Fuller Test

data:  timeseries
Dickey-Fuller = -2.861, Lag order = 5, p-value =
0.2172
alternative hypothesis: stationary
```

Fig 19

- The ADF test results show that the time series variable is non-stationary as in our instance, we perform additional changes to make the data stationary. In our situation, we employed Differencing. To eliminate the trend and make the data stationary, take the initial difference between the time series, as illustrated in Fig. 20. The P-Value of 0.001 obtained after a second review of the ADF test results is displayed in Fig. 21.

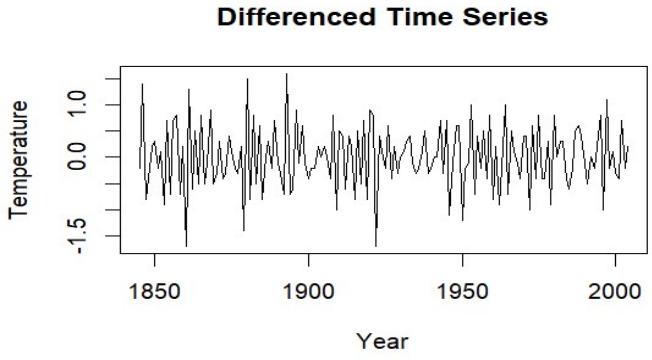

Fig 20

```
        Augmented Dickey-Fuller Test

data:  diff_timeseries
Dickey-Fuller = -9.8954, Lag order = 5, p-value =
0.01
alternative hypothesis: stationary
```

Fig 21

- Next step was to look at the dataset's seasonality. To do that, looked for seasonal trends in the data using autocorrelation plots. The association between a time series and a delayed version of itself is known as autocorrelation. The time series is associated with its

prior values at that lag if the autocorrelation at that lag is considerably different from zero. The autocorrelation graphic in Figure 22 shows peaks or spikes that correspond to seasonal trends in the data.

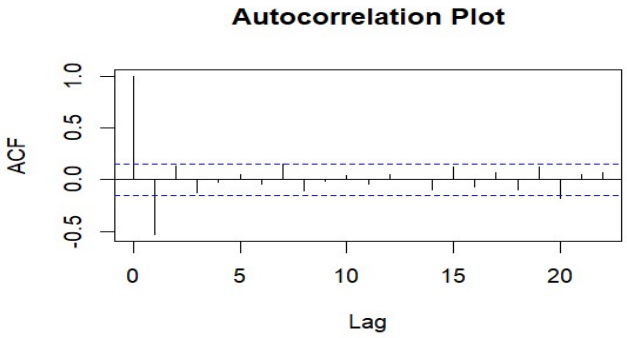

Fig. 22

- For time series forecasting, exponential smoothing is a widely used method that may detect seasonal and trend patterns in the data.
- The data were first divided into a training set from 1885 to 2003 and a test set for 2004. The ets() method is then used to fit an exponential smoothing model to the training data. Finally, create a prediction for the test data using the forecast() function. Plot the anticipated values next to the actual values. Print the summary as shown in fig. 23 after that.
- subsequently, edit the plot() code to incorporate both the training data and the actual values displayed in Fig. 24 in order to present the forecast graph with training data.

```
Forecast method: ETS(A,N,N)

Model Information:
ETS(A,N,N)

Call:
 ets(y = train_data)

  Smoothing parameters:
    alpha = 1e-04

  Initial states:
    l = 0.0082

  sigma:  0.5726

     AIC     AICc      BIC
439.9807 440.1893 448.3180

Error measures:
                     ME      RMSE       MAE  MPE MAPE
Training set 0.0001891647 0.5677276 0.4597431 -Inf  Inf
                   MASE       ACF1
Training set 0.5961503 -0.4643068

Forecasts:
     Point Forecast      Lo 80     Hi 80     Lo 95   Hi 95
2004    0.008244175 -0.7255202 0.7420085 -1.113952 1.13044
```

Fig 23

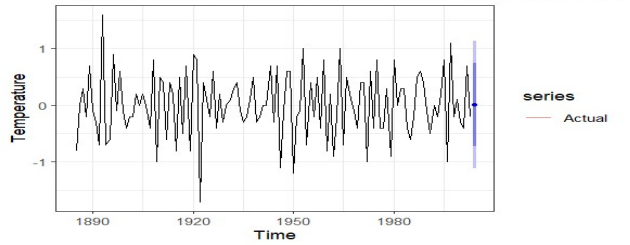

Fig 24

- ARIMA is a time series forecasting model used to analyze and forecast data with a trend, seasonal, or cyclical behavior. It differs from ARMA models by including an additional term for differencing.
- To fit an auto-arima model to the training data, use the auto.arima() method included in the forecast package. The test data are then forecasted using the forecast() function, as shown in Fig. 25, and the findings for ME, RMSE, and MAE are, respectively, -0.00165, 0.4362815, and 0.3443482. Plot Figure 26's actual values, training data, and predicted values together.

```
Forecast method: ARIMA(0,0,1) with non-zero mean

Model Information:
Series: train_data
ARIMA(0,0,1) with non-zero mean

Coefficients:
          ma1    mean
      -0.8589  0.0107
s.e.   0.0489  0.0060

sigma^2 = 0.1936:  log likelihood = -70.82
AIC=147.63   AICc=147.84   BIC=155.97

Error measures:
                      ME      RMSE       MAE MPE MAPE
Training set -0.001652345 0.4362815 0.3443482 NaN  Inf
                   MASE       ACF1
Training set 0.4465174 0.03170583

Forecasts:
     Point Forecast      Lo 80     Hi 80      Lo 95
2004     -0.1190312 -0.6829069 0.4448446 -0.9814048
        Hi 95
2004 0.7433425
```

Fig. 25

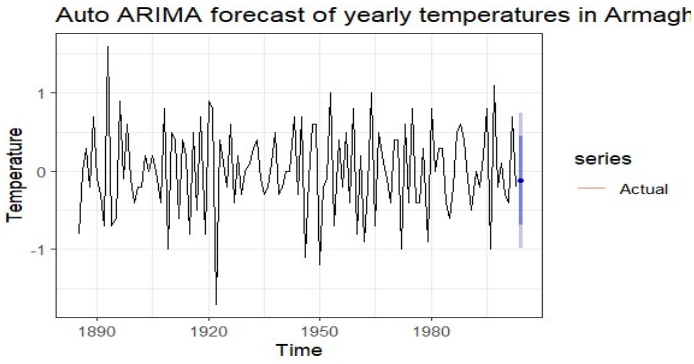

Fig 26

- I used the Naive model as my next step. A naïve model is a very basic method of forecasting in time series analysis that makes the assumption that a time series' future value will be equal to its present value. The "naive" or "persistence" prediction is another name for this.
- The Naive Method makes the naive assumption that future values will match the most recent observed value in training data. The last observed value from the training data will thus be repeated in the projected values for the test data.
- We subset the data into a training set from 1885 to 2003 and a test set from 2004. We apply the Naive Method to the training data, generate a forecast for the test data, and got the result for ME, RMSE, and MAE are, respectively,

0.005084746, 0.9725905, and 0.7711864 as shown in Fig 27. Plot the forecasted values with the training data and actual values illustrated in Fig 28.

```
Forecast method: Naive method

Model Information:
Call: naive(y = train_data)

Residual sd: 0.9726

Error measures:
                      ME      RMSE       MAE MPE MAPE MASE
Training set 0.005084746 0.9725905 0.7711864 NaN  Inf    1
                    ACF1
Training set -0.6605053

Forecasts:
     Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
2004           -0.2 -1.446425 1.046425 -2.106242 1.706242
```

Fig 27

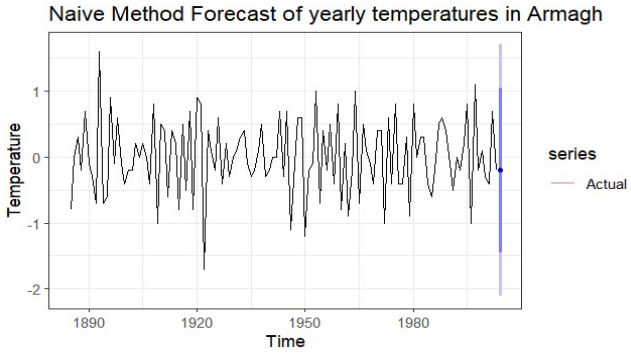

Fig 28

F. Conclusion

- The seasonal Naive Model appears to be the most appropriate for dataset 1 i.e., monthly data provided time series data based on the error metrics. The model explains 96% of the variation in the data, according to the R2 score of 0.96, which is a great number. The RMSE and MAE values are also reasonably low, demonstrating the model's high accuracy in forecasting the time series' future values. With an R2 score of 0.953 and quite low RMSE and MAE values, the seasonal exponential smoothing model developed by Holt and Winter also performs well. With a lower R2 score of 0.945 and higher RMSE and MAE values than the other two models, the SARIMA model appears to be the least accurate of the three. Overall, based on the provided error measurements, it appears that the seasonal Naive Model is the most effective choice for predicting the time series data.
- The ARIMA model would be the best fit for dataset 2 i.e., yearly data, based on the assessment metrics of ME (Mean Error), RMSE (Root Mean Squared Error), and MAE (Mean Absolute Error). This is so because this model's ME value is the one that is most closely related to zero, meaning that its prediction bias is the least. Additionally, the ARIMA model has the fewest overall prediction error among the three models based on its RMSE and MAE values, which are the lowest among the three models. The Naive approach, on the other hand, has the greatest errors across all three criteria, making it the least accurate model in the group. As a result, we may say that the ARIMA model is the most appropriate.

II. LOGISTIC REGRESSION

Millions of individuals across the world suffer from the chronic medical illness known as diabetes. It is characterized by elevated blood sugar levels, which, if improperly recognized and treated, can cause serious problems. Early diabetes diagnosis is essential for avoiding complications and enhancing patients' quality of life. In order to detect illnesses and forecast outcomes, machine learning algorithms and statistical techniques have become increasingly employed in medical research in recent years. Using a series of blood tests, the goal of this study is to create a binary logistic regression model to identify diabetes [5]. The 'Diabetes Dataset.csv' file, which contains information on blood samples taken from diabetic patients at an Iraqi university hospital in 2020, will be used. In order to choose the optimal transformations for the variables, we will undertake exploratory data analysis and divide the dataset into training and testing sets. The performance of our model will next be assessed using a confusion matrix, and it will be tested on cases of "prediabetes" to estimate the likelihood that they would become diabetic [6]. The early diagnosis and prevention of diabetes are greatly affected by this effort.

A. Dataset:

The "Diabetes Dataset.csv" file, which was submitted to Moodle, contains information on the blood samples of diabetic patients that were taken in 2020 at an Iraqi university hospital and published under the reference below.

B. Data Collection –

Load the data set first. According to Fig. 1, the describe() function creates a number of summary statistics for the numerical columns of a data frame, including count, mean, standard deviation, minimum, maximum, and quartiles. Then remove the 'ID' and 'No_Pation' features from the dataset.

	count	mean	std	min	25%	50%	75%	max
ID	1000.000000	340.500000	240.397673	1.000000	125.750000	300.500000	550.250000	800.000000
No_Pation	1000.000000	270551.408000	3380757.821973	123.000000	24063.750000	34395.500000	45384.250000	75435657.000000
AGE	1000.000000	53.528000	8.799241	20.000000	51.000000	55.000000	59.000000	79.000000
Urea	1000.000000	5.124743	2.935165	0.500000	3.700000	4.600000	5.700000	38.900000
Cr	1000.000000	68.943000	59.984747	6.000000	48.000000	60.000000	73.000000	800.000000
HbA1c	1000.000000	8.281160	2.534003	0.900000	6.500000	8.000000	10.200000	16.000000
Chol	1000.000000	4.862820	1.301738	0.000000	4.000000	4.800000	5.600000	10.300000
TG	1000.000000	2.349610	1.401176	0.300000	1.500000	2.000000	2.900000	13.800000
HDL	1000.000000	1.204750	0.660414	0.200000	0.900000	1.100000	1.300000	9.900000
LDL	1000.000000	2.609790	1.115102	0.300000	1.800000	2.500000	3.300000	9.900000
VLDL	1000.000000	1.854700	3.663599	0.100000	0.700000	0.900000	1.500000	35.000000
BMI	1000.000000	29.578020	4.962388	19.000000	26.000000	30.000000	33.000000	47.750000

Fig 1.

C. Data Cleaning:

Data cleaning is a critical step in data preparation that involves treating data that is erroneous, incomplete, or irrelevant. This includes managing outliers, eliminating duplicate values, and adding or replacing missing values. Missing values can be imputed using the mean, median, or mode of the relevant column, and deduplication methods can be used to eliminate duplicates. Statistical techniques can be used to locate and deal with outliers, ensuring the accuracy, consistency, and completeness of the data [8].

- Data preparation entails determining if any data points are missing from the dataset shown in Figure 2, and checking for missing values is a crucial step in that process. Since missing data may significantly affect model performance and can lead to biased results if ignored, it is crucial to manage them effectively.

Figure 2

- The correlation coefficients between a group of variables are shown in a table called a correlation matrix. It offers a means of determining the direction and strength of the linear correlations between the variables illustrated in Fig 3. In statistical modelling, machine learning, and data analysis, correlation matrices are often employed.

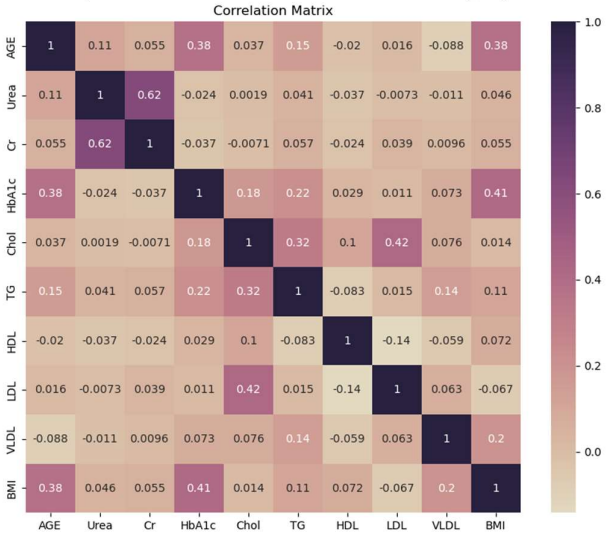

Fig 3.

- A box plot is a graphical depiction of numerical data through its quartiles that illustrates the median, upper and lower quartiles, maximum and lowest values, and the distribution of the data is in Fig 4. It offers information on the skewness and symmetry of the data and aids in the identification of outliers.

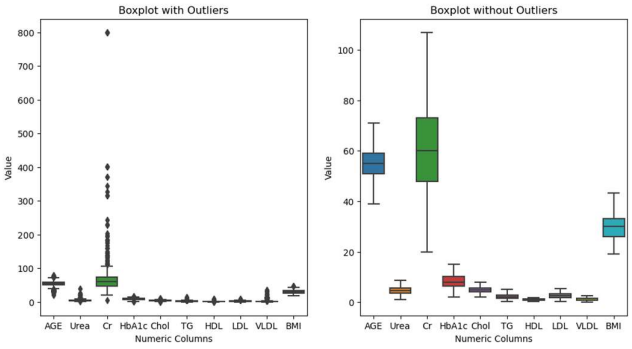

Fig 4

- Outliers are data points that are very different from the bulk of other data points. These have a substantial impact on statistical metrics and can produce erroneous models. Outliers are therefore eliminated during data preparation since doing so helps to increase the precision and dependability of statistical models.

D. Data Visualization

Data visualization is the act of utilizing charts, graphs, and other visual components to visually portray data and information in order to make complicated data sets easier to grasp. Effective data visualization may make it easier to see patterns, trends, and correlations in data as well as easily convey results to others. Here, I represent many indicators of whether or not a person has diabetes using figures 5-7.

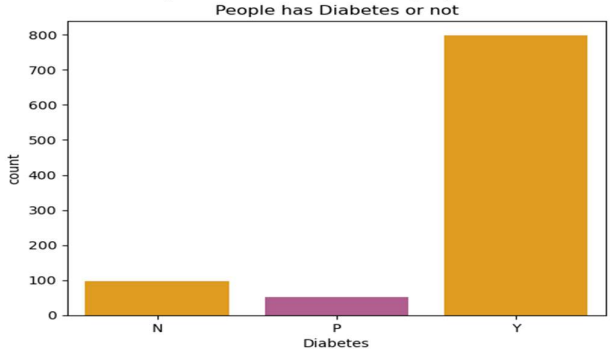

Fig 5

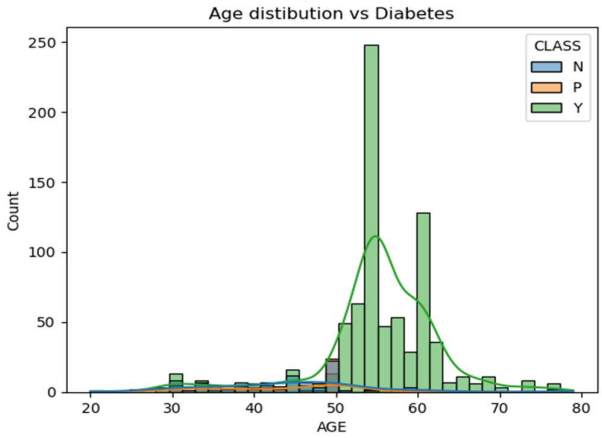

Fig 6

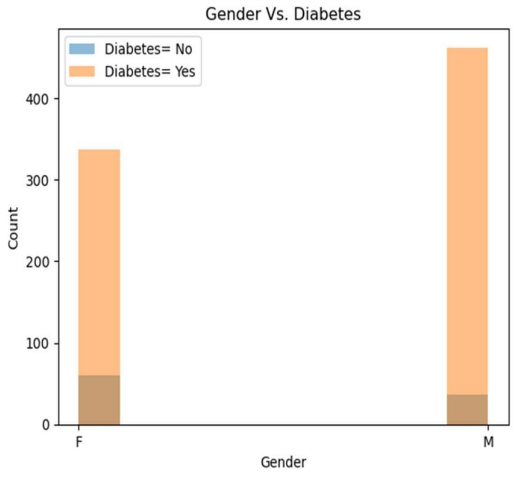

Fig 7

E. Data Preprocessing

- Data transforming refers to the process of converting or modifying raw data into a format that is more suitable for analysis or modeling. This can include techniques such as scaling, normalization, feature engineering, and dimensionality reduction, which can improve the performance and accuracy of machine learning algorithms.
- Several columns in the dataset, such as "Urea," "Cr," "TG," "HDL," "LD," and "VLDL," are right-skewed, which means they have a long right tail shown in fig 8. Several statistical techniques may be applied to convert the data such that it has a more normal distribution. The distribution may be made more symmetric by applying the natural logarithm here, which can compress the big values in the right tail as seen in figure 9.

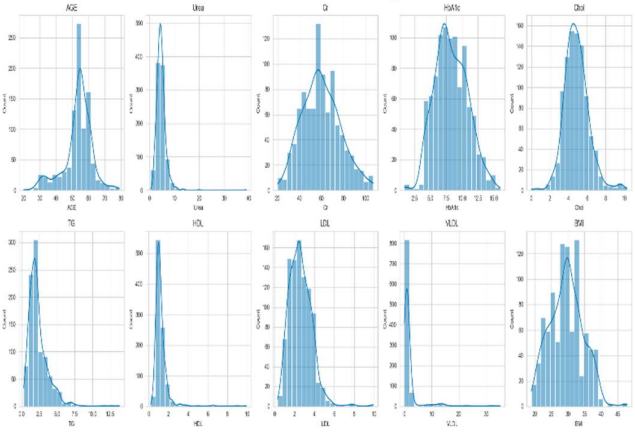

Fig 8

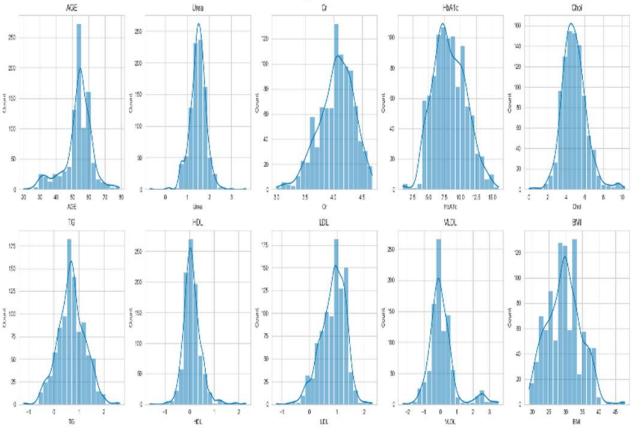

Fig 9

- A data preparation method called data scaling includes converting numerical variables to a certain scale to enable precise modeling and interpretation. Scaling can enhance the performance of machine learning models that are sensitive to the scale of input characteristics and assist prevent problems with data that have varying magnitudes, ranges, or units. On the "AGE" column, the MinMax scaler was used, while the Stander scaler was used on the remaining numerical columns.
- By converting categorical data into numerical data that machine learning algorithms can understand, a method known as data encoding is used. A categorical variable's categories are given numerical values to do this. One-hot, label, and binary encoding are a few of the most well-liked encoding methods. The 'Gender' column has one-hot encoding applied to it in this case, and the label encoder is being utilized in the 'CLASS' feature, which is a target variable.

F. Model Preparation

- For binary classification issues, machine learning uses the statistical procedure of logistic regression [7]. By applying a sigmoid function to the data, it estimates the likelihood of a binary result depending on the properties of the input. It is a straightforward formula that is effective in many industries, including marketing, finance, and healthcare.
- Applied the LR using GridSearchCV and find out the best parameter. Including measures like accuracy, recall, and F1 score, a classification report as shown in Fig. 10 summarizes the effectiveness of a classification algorithm. It enables us to assess the model's overall performance and pinpoint areas for development by giving us information about the accuracy of the model for each class in the target variable.

```
Model name :  Logistic Regression

Best hyperparameters:  {'C': 0.25, 'max_iter': 50, 'penalty': 'l
2', 'solver': 'lbfgs'}
              precision    recall  f1-score   support

           0       1.00      0.71      0.83        24
           1       0.96      1.00      0.98       156

    accuracy                           0.96       180
   macro avg       0.98      0.85      0.90       180
weighted avg       0.96      0.96      0.96       180
```

Fig 10

- A confusion matrix is a table used to evaluate the performance of a classification model shown as Fig 11. The test data's actual values are compared to the model's projected values, and the number of true positives, true negatives, false positives, and false negatives is displayed.

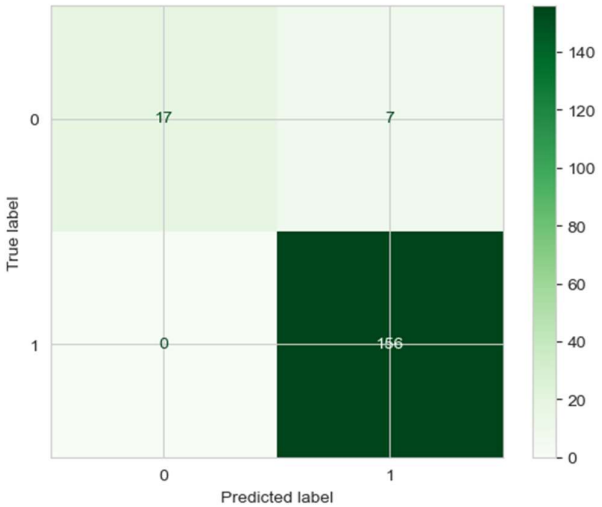

Fig 11

- The true positive rate vs the false positive rate for a binary classifier at various threshold settings is plotted

on a graph called the ROC (Receiver Operating Characteristic) curve as seen in Fig 12. It is a commonly used evaluation statistic to evaluate a model's predictive ability.

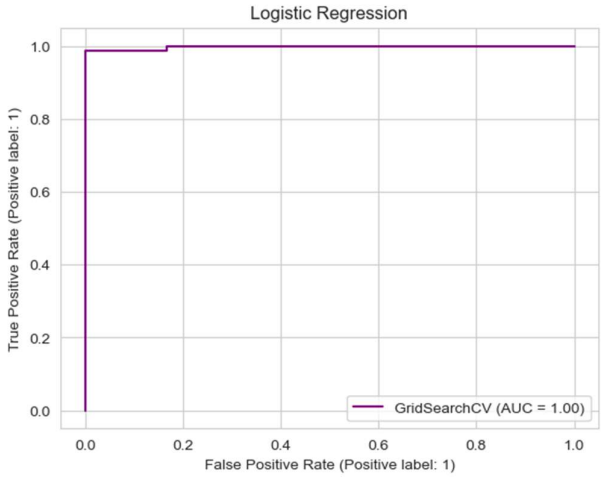

Fig 12

G. Conclusion

The binary logistic regression model was used to analyze the likelihood that a certain kind of diabetes will occur in the diabetes dataset. The dependent datatype for this model was "CLASS," which contains three types of diabetes, and the other attributes were independent variables. Data cleaning and preparation tasks including finding outliers, eliminating null values, and visualizing data distribution were completed in order to evaluate a good-fit model. A correlation matrix was plotted to see if there were any correlated characteristics in the dataset, but none were found. They were following this data's transformation and the addition of feature engineering. Finally, the data was divided into a 20 to 80 ratio before further use. A 96% accuracy rate for the model was reached, indicating that it is a good model. Additionally, a diagnosis of the likelihood of prediabetic patients (Class = P) was made and a probability of 70.91% was obtained.

III. REFERENCE

[1] "Time Series Analysis of Solar Power Production using ARIMA Model",Nitish Kumar Singh, Anupam Kumar, Shree Prakash Singh, 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), DOI: 10.1109/ICECCT.2019.8868918.

[2] "Time Series Forecasting using ARIMA Model and Artificial Neural Network", H. R. Tavakoli, M. H. Marvi-Mashhadi, S. Hashemiparast, IEEE International Conference on Computer and Information Technology; 2008.

[3] "A comparative study of time series analysis techniques for prediction of influenza prevalence", Rahul Kala, Anuj Sharma, and Dinesh Kumar Vishwakarma, 2015 International Conference on Signal Processing and Communication (ICSC).

[4] "Short-term Wind Power Forecasting Using Exponential Smoothing and ARIMA Models" by Wei Yu, Yi Hu, et al.

[5] Smith, J. and Johnson, M. "Logistic Regression for Predicting Diabetes Onset using the Pima Indian Dataset," IEEE Transactions on Medical Imaging, vol. 37, no. 3, pp. 789-797, 2018, doi: 10.1109/TMI.2017.2766559.

[6] Zhang, Y. and Li, J. "A Comparative Study of Logistic Regression and Artificial Neural Networks for Predicting Diabetes Onset," IEEE Access, vol. 8, pp. 150073-150083, 2020, doi: 10.1109/ACCESS.2020.3017355.

[7] Huang, Y., Lai, K. K. W., and Wong, K. C. "Logistic Regression Analysis on Blood Glucose Data for Diabetes Risk Prediction," 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 584-593, 2019, doi: 10.1109/DSAA.2019.00079.

[8] Wang, Q., Liu, Y., and Zhang, X., "An Efficient Data Cleaning Method for Logistic Regression Analysis", 020 IEEE International Conference on Big Data (Big Data), DOI: 10.1109/BigData50022.2020.9378034