Statistics for Data Analytics

Terminal Assessment Report on
Time Series Analysis and Logistic Regression

Debayan Biswas
Student ID: 22242821
Master of Science in Data Analytics
National College of Ireland
x22242821@student.ncirl.ie

***Abstract*—This project encompasses a dual analysis, including Time Series Analysis on historical weather data and Binary Logistic Regression on a dataset related to cardiac conditions. The objective is to estimate suitable time series models and logistic regression models, respectively, followed by rigorous evaluation and interpretation of the results.**

I. Introduction

A. Time Series Analysis

The Time Series Analysis involves the exploration of historical weather data from Met Eireann's Dublin Airport weather station, provided in comma-separated value format. The data set consists of 9 features and several rows. The analysis focuses on a specific variable, aiming to estimate suitable models for further prediction and forecasting. The time-series analysis will be carried out on the variable 'maxtp - Maximum Air Temperature (C)' which is as per the last digit of my student number which is x22242821.

B. Logistic Regression Analysis

The cardiac dataset contains attributes of 100 participants, in comma-separated value format. The goal is to estimate a binary logistic regression model to understand the relationships between various factors and how these factors influence the presence or absence of cardiac conditions in participants. Descriptive statistics, visualizations, and dimensionality-reduction techniques will be employed to enhance understanding of the dataset.

II. Time Series Analysis

A. Exploratory Data Analysis

1) Data set Summary:
The weather dataset provided for the analysis has 29889 rows and 9 features in total, having 3 object type columns which also include a 'date' column, and the rest of the columns have integer type datatype, mostly float type data. On further investigation, it can be stated that the two object-type data columns other than the 'date' column have numeric data in them. They are considered objects due to the presence of continuous data and might have a presence of Blank spaces in them which will be investigated in the later sections. The summary of the dataset is shown in Fig 1.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29889 entries, 0 to 29888
Data columns (total 9 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   date                                          29889 non-null  object
 1   maxtp(Maximum Air Temperature - degrees C)    29889 non-null  float64
 2   mintp(Minimum Air Temperature - degrees C)    29889 non-null  float64
 3   gmin(Grass Minimum Temperature - degrees C)   29889 non-null  object
 4   rain(Precipitation Amount - mm)               29889 non-null  float64
 5   cbl (Mean CBL Pressure-hpa)                   29889 non-null  float64
 6   wdsp(Mean Wind Speed - knot)                  29889 non-null  float64
 7   pe(Potential Evapotranspiration - mm)         29889 non-null  float64
 8   evap(Evaporation -mm)                         29889 non-null  object
dtypes: float64(6), object(3)
memory usage: 2.1+ MB
None
```

Fig. 1: Summary of dataset

2) Unique Values:
The data set has unique values present in all the columns with nominal repetition of data. However, two columns, 'gmin' and 'evap' have continuous incrementing data which might be considered categorical in some cases. Due to the immense number of data present in each of the columns and with the presence of very high uniqueness, they are not considered categorical.

3) Missing Data:
The blank or Null values are initially checked using 'isna' function but it returns no value. The reason for not finding a null or blank space is that the two columns that consist of blank spaces are object-type columns. Thus 'applymap' function, which is applied to every element of the dataset, is implemented to detect and remove the unnecessary blank values by dropping rows with blank spaces. In total 7 blank rows from both the object type data columns (other than the 'date' column) are removed and the row count changes to 29882.

4) Preparing Date Columns for Time-related operations:

a) Converting object type date column to date type:
The 'date' column present is the most significant column for implementing the time-series analysis on the given dataset. However, the column being an object-type column will not allow for any exploration or analysis of data due to possible data type mismatch. To overcome this situation, the 'date' column is converted to a date-time type column.

b) Extracting day, week, month from date: The date column is now further expanded into day, week, month, and

year columns for visualization of data. The weather report for a particular interval can be visualized with the help of the expanded table data which is shown in details in the next section.

5) Visual Exploration:

a) Yearly Aggregation of Average Air Temperature:

A simple time-series plot is visualized by using the extracted column 'year' and the variable in consideration for analysis which is maximum air temperature. The plot in Fig 2. shows an increasing maximum temperature trend over the years from around 1942 to 2023, with a sudden peak after 2020 as the global temperature is rising drastically as compared to a decade ago.

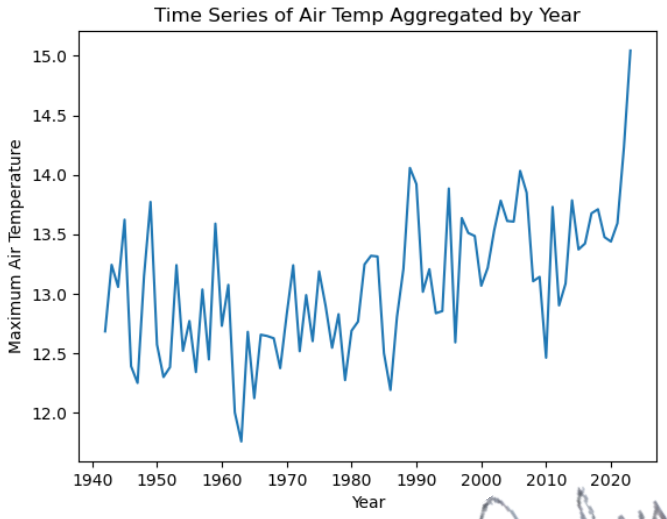


Fig. 2: Plot for Yearly Aggregation of Average Air Temperature

b) Year-wise Monthly Average Temperature:

The maximum temperature feature is plotted against the extracted year column which is grouped by Month. The spikes indicate a sudden increase in the Maximum temperature for a given month of a year thus depicting the presence of seasonality in the dataset as it repeats every year.

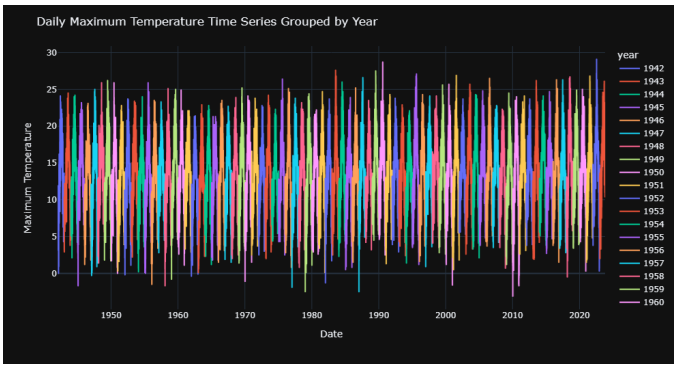


Fig. 3: Daily Maximum Temperature Time Series Grouped by Year

c) Daily Maximum Temperature Time Series Grouped by Year:

The daily maximum temperature of the time series data as shown in Fig.3 is grouped by year. This plot shows the daily maximum temperature for every year in the dataset. The plot is plotted using the Plotly function and can be zoomed in or out as per convenience to check the details breakdown of the recorded daily data.

d) Time-series decomposition:

The time-series decomposition in Fig. 4 shows comprehensive details of the dataset. The seasonal trend for the year range is depicted followed by trend and seasonal data. A similar repeated trend can be noticed for the whole interval suggesting the data has a seasonal repetition pattern. The figure also shows the residuals for the given year range.

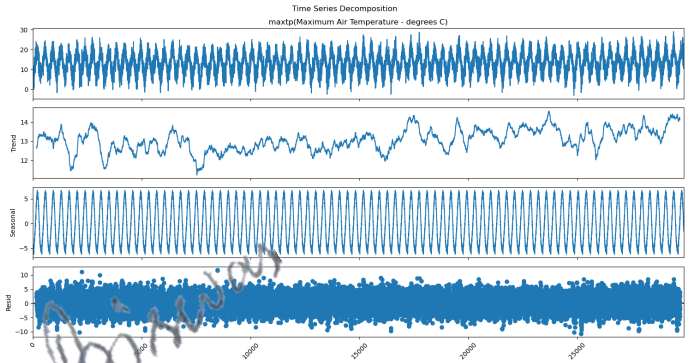


Fig. 4: Time-series decomposition

B. Data Preparation

1) Converting Object Type data to numeric type:

The two columns, 'gmin' and 'evap' holds object-type data in the dataset provided. However, from a close examination done on the above sections, it was noticed that the columns were being considered as object type as they have continuous numerical data and also consist of blank values which have been removed in the previous sections using appropriate technique. The values of this column are numeric and have significance in the analysis of data. The type conversion of these two columns is computed and the object type changes to numeric float type data.

2) Outliers detection and removal:

The box plot is implemented on the dataset for identifying outliers and the spread of the data on all the numeric data columns. Outliers are present in the data set as can be seen in Fig.5. The outliers can influence the outcome of the prediction and it needs to be removed from the dataset. On observation of the boxplot, it is evident that some of the columns do have outliers, especially the column 'rain'. The outlier percentage is also checked for all the numeric features and the same column 'rain' consists of a high outlier percentage.

Z Score of numeric features is calculated. The z score threshold is taken as 2 for this data set, and the rows having a z score above the mentioned threshold are to be removed to achieve

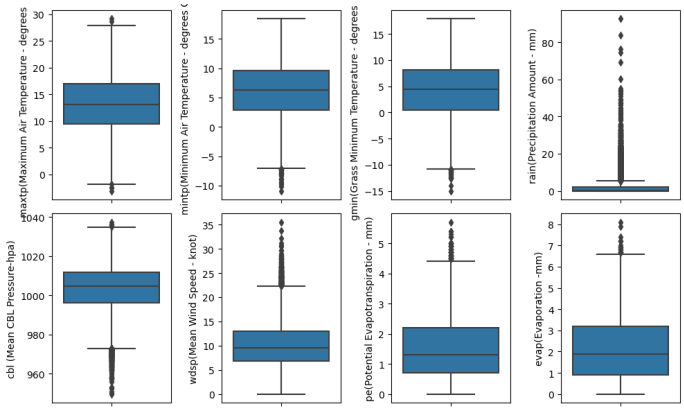


Fig. 5: Outlier before Removal

clean data. After removing the outliers, the row count falls to 23620.

The boxplot for the cleaned data is again checked which can be seen in Fig.6. From the boxplot, it can be observed that the outliers of all the columns are removed except for the column 'rain' which still has a significant number of outliers present. So the column 'rain' doesn't hold any significant value that might impact the analysis of the data, and also it is not a considerable feature here so it can be neglected.

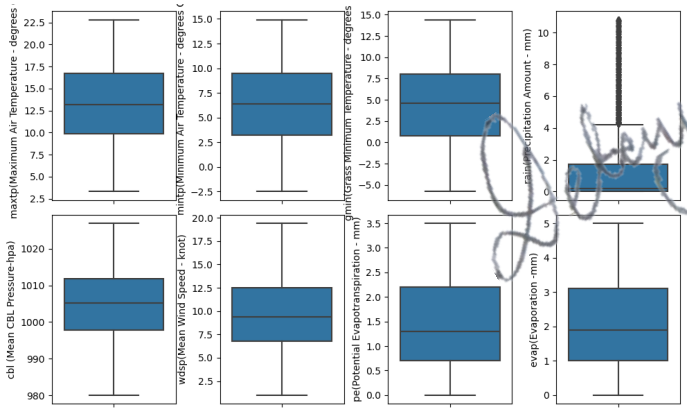


Fig. 6: Outlier after Removal

C. Modeling

1) Splitting of Data:

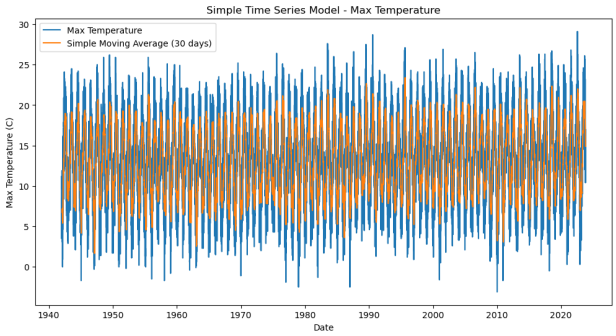
The cleaned data set should be split into train and test sets. The train set will be used to train the data model that will be implemented and the test set will be used to test the trained model and will be used for forecasting. The data from 2019 to 2022 will be used to fit the models and data from 2023 will be used to evaluate the performance of the fitted. Before splitting, the date column is formatted as per the required format of day month, and year.

The cleaned data is now split into train and test sets which hold 1168 and 226 rows respectively.

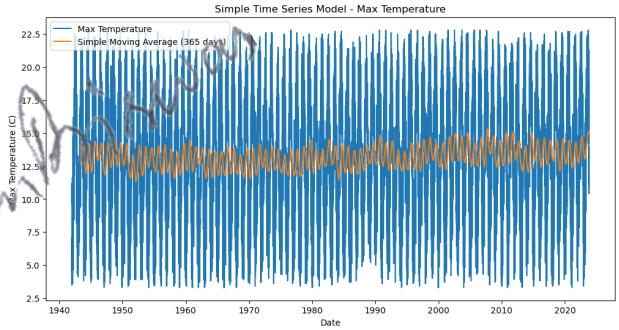
2) Model Implementation:

a) Simple Time Series Model:

The simple moving average (SMA) is a time series model used for time series forecasting. It plots the average value of a set of data points over a specified period in time, where each data contributes equally to the average and hence the name is moving average. The SMA plot in Fig.7(a) plots the Monthly moving average of the maximum temperature feature from the year 1942 to 2023. The plot in Fig.7(b) shows the yearly moving average for the same period. Both the plot shows the presence of seasonality. A simple time series plot is



(a) SMA for Monthly basis



(b) SMA for Yearly basis

Fig. 7: Simple Moving Average plot

also implemented on the dataset for the maximum temperature column. The model is trained first using the train data set and then fitted into the test data set. The result shows the same seasonal pattern in the test data with a more detailed view which can be observed in Fig.8.

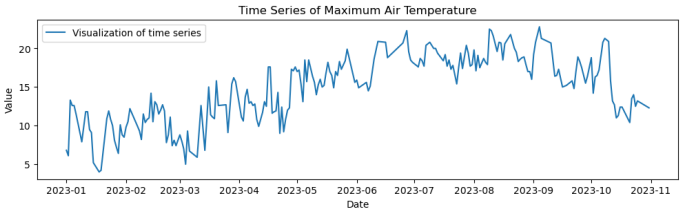


Fig. 8: Simple Time-Series plot

b) Exponential Smoothing:

Exponential Smoothing is a time series forecasting and prediction method that assigns weights to past observations mainly decreasing weights. It is designed to adapt quickly

to changes in data while providing a proper representation of trends and seasonality. Exponential smoothing has different variations but Triple Exponential Smoothing, also known as Holt-Winters Method has been implemented here. This model is implemented to incorporate seasonality as the dataset itself is weather data from Met Eirin which is by itself seasonal and repeats every year. Here the 'trend' and 'seasonal' parameters indicate an additive seasonality. This implies that the trend and seasonality components are added together, making it an additive model. Also, the seasonal period is set to 365, suggesting that the model assumes a yearly seasonality pattern. The pattern and trend are discernible in Fig.9, depicting the observed training data along with the actual and predicted test data. The forecast data follows a similar trend to the

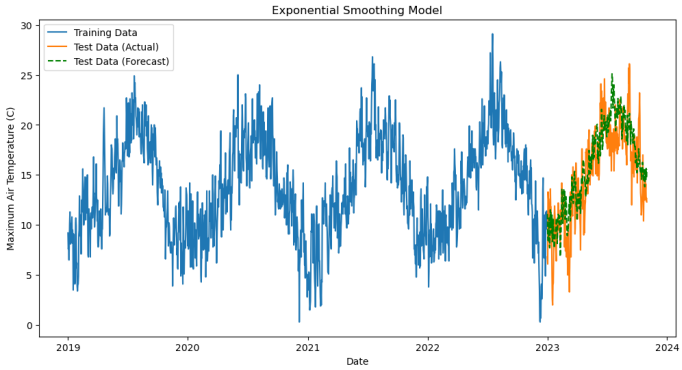


Fig. 9: Exponential smoothing

test data suggesting that the forecast is properly implemented. Exponential Smoothing strikes a balance between simplicity and accuracy. The choice of smoothing parameters is crucial and depends on the specific characteristics of the data and the desired responsiveness to changes.

c) ARIMA and SARIMA:

ARIMA (AutoRegressive Integrated Moving Average) and SARIMA (Seasonal AutoRegressive Integrated Moving Average) are two commonly employed time series forecasting models for predicting future values using historical data. Both models are auto-regressive in nature, meaning they make predictions about future behavior based on past data patterns. 1. ARIMA is a time series forecasting model that combines auto-regressive, differencing, and moving averages. It captures the relationship between present and past data, ensures stationarity through differencing, and models short-term fluctuations. ARIMA employs parameters (p, d, q), where p represents the order of the autoregressive components, d is the degree of differencing, and q denotes the order of the moving average component. ARIMA is not effective for seasonal data, yet the ARIMA model has been implemented in the dataset to check for the result which is depicted in Fig.10. The AIC (Akaike's Information Criterion), utilized as a criterion for model selection from a finite set of models, is applied in this context to assess the optimal value and determine the best order of p, d, q for the model. The best parameter obtained gives a forecast in the form of a straight horizontal line depicting that ARIMA is not efficient in finding the seasonal forecasting of data.

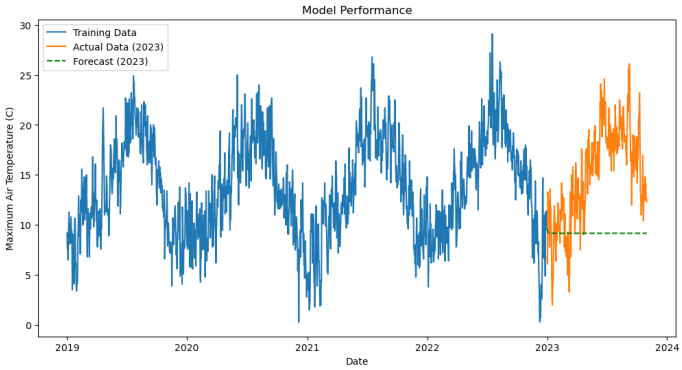


Fig. 10: ARIMA MODEL

2. SARIMA is a time series model that further extends ARIMA to add a seasonal component for handling time series data with clear and proper seasonality. SARIMA uses(p, d, q) and (P, D, Q, m), where (p, d, q) are non-seasonal components which are the same as done previously with ARIMA, (P, D, Q) are the seasonal components, and m (also denoted by s) is the number of observations per season. SARIMA is suitable for forecasting when there is a noticeable seasonal trend in the data. Then train data for the given years $2019-01-01$ to $2022-12-31$ is passed into the model. Then the test data for the period $2023-01-01$ to $2023-10-31$ is fitted into the train model.

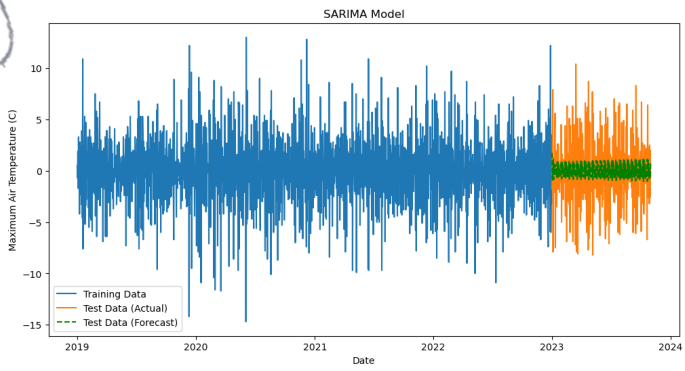


Fig. 11: SARIMA MODEL

Fig.11 shows the graph for the SARIMA model incorporated. From the observation, it can be inferred that this model is giving proper forecasting into the future for the seasonal trend in the data. The plot in Fig.12 shows the actual test and forecast test data. The forecast shows a similar trend thus depicting proper forecasting.

D. Diagnostics

To verify and ensure the validity and reliability of the SARIMA model, diagnostic tests such as the Ljung-Box test for residual autocorrelation and the Jarque-Bera test for the

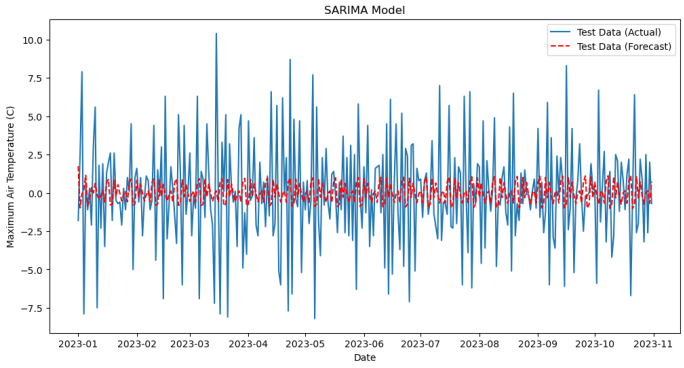


Fig. 12: SARIMA MODEL Forecast

normality of residuals are performed.

Model Diagnostics:

SARIMAX Results

Dep. Variable:	maxtp(Maximum Air Temperature - degrees C)	No. Observations:	1461
Model:	SARIMAX(1, 1, 1)x(0, 2, [1, 2, 3], 12)	Log Likelihood	-3739.146
Date:	Sun, 31 Dec 2023	AIC	7490.292
Time:	00:56:34	BIC	7521.910
Sample:	01-01-2019	HQIC	7502.097
	- 12-31-2022		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.5472	0.021	-25.746	0.000	-0.589	-0.506
ma.L1	-0.9996	0.533	-1.876	0.061	-2.044	0.045
ma.S.L12	-2.0251	0.046	-43.886	0.000	-2.116	-1.935
ma.S.L24	1.0649	0.067	15.893	0.000	0.934	1.196
ma.S.L36	-0.0391	0.028	-1.387	0.165	-0.094	0.016
sigma2	9.3492	4.959	1.885	0.059	-0.370	19.069

Ljung-Box (L1) (Q):	62.55	Jarque-Bera (JB):	23.57
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	0.92	Skew:	-0.23
Prob(H) (two-sided):	0.34	Kurtosis:	3.43

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

Ljung-Box Test for Residuals Autocorrelation:
Ljung-Box Test Statistic: lb_stat
P-value: lb_pvalue

Fig. 13: SARIMA MODEL SUMMARY

The low Prob(Q) in the Ljung-Box test implies autocorrelation in residuals, and the low Prob(JB) in the Jarque-Bera test indicates non-normality in residuals. The autocorrelation can be minimized to a certain extent by increasing the lag order but that also depends on the type of data that has been used. The log transform might fix the normality issue among the residuals. From the model summary in Fig.13 ,it can be observed that the AIC and BIC values are 7940 and 7521 respectively.

E. Evaluation

1) Exponential Smoothing: The Exponential Smoothing model's performance is evaluated with the help of some classification metrics. The root mean squared error(RSME) is evaluated to quantify the model's efficiency, which stands at 3.03 which is a fairly low and desirable score. The outlier sensitive mean squared error value is 9.18 and the absolute mean error is at 2.42. The three metrics indicate better model accuracy.

2) SARIMA: The SARIMA model's performance is evaluated using the test dataset. Forecasted values are compared with actual observations, and the root mean squared error (RMSE) is calculated to predict the accuracy of the model. The RMSE value of 3.495 is fairly low and desirable. ARIMA is not considered as it doesn't provide a proper analysis of the seasonal data and also has a high RSME of 7.56. The Mean Squared Error (MSE) for SARIMA is 12.22 whose value is sensitive to outliers, as it is a square of the errors. A lower MSE is generally better and this value of MSE is not high suggesting it is a decent value.

III. Logistic Regression Analysis

A. Exploratory Data Analysis

1) Data set Summary:
The data set provided has 100 rows and 6 features in total having 'gender' and 'cardiac_condition' as object type datatype columns. The Gender type is depicted as Object type as it has textual data indicating either male or female genders. The cardiac condition column holds binary data in a textual format as present or absent, which denotes the presence or absence of the cardiac condition in the patient. These object type data types need to be encoded for better understanding and analysis of the data.

2) Unique Values:
The dataset consists of unique values in all the columns except for the two object-type columns as they are considered categorical. The column named 'caseno' doesn't hold any relevant value for the analysis and hence it is dropped from the table.

3) Missing Values:
The cardiac dataset doesn't hold any missing values on it. The dataset doesn't hold any Null or blank value which may hamper the operation and the dataset is completely clean.

B. Data Preparation and Visual Exploration

1) Encoding necessary features:

a) One-Hot Encoding Method:
The gender column can't be encoded with label encoding as the column will fetch binary values after encoding which are 0 and 1, which is not desirable for analysis. One-hot encoding is used because gender is a nominal variable without a meaningful ordinal relationship. One-hot encoding ensures that the model does not interpret the numerical labels as having inherent order. So to avoid this problem, the gender column is encoded using One-Hot encoding which generates two new binary columns for both genders, male and female.

b) Label Encoding the target column "cardiac_condition":
The target variable which is the cardiac_condition column is present in binary format. The target or the dependent column represents a binary class, depicting features 'present' or 'absent'. For implementing further operation on the column, it is necessary to encode the column into numeric format. However, many machine learning algorithms are capable of handling binary classes directly without the need for label encoding. Still, the data column is converted into a numeric column for better usability. The column after implementation

of label encoding gives the result in 0 and 1 numeric format representing the cardiac condition as either 'absent' or 'present'. The target variable distribution in Fig.14 shows the

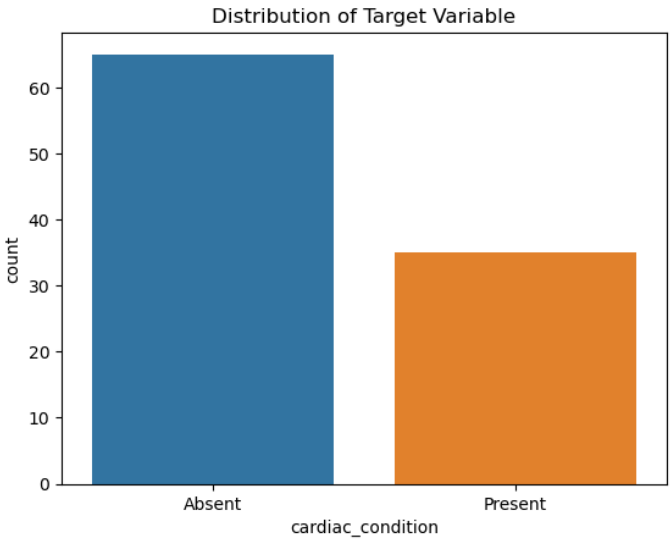


Fig. 14: Distribution of Cardiac Condition

distribution of cardiac condition whether it is present or absent.

2) Correlation among data:

The correlation among the individual features in the dataset is checked. From the observation of the Fig.15, it is evident that the correlation matrix doesn't have more of a correlation among its members. Thus further operations of removing correlation and transformation are not necessary for the dataset. The correlation value for the individual columns

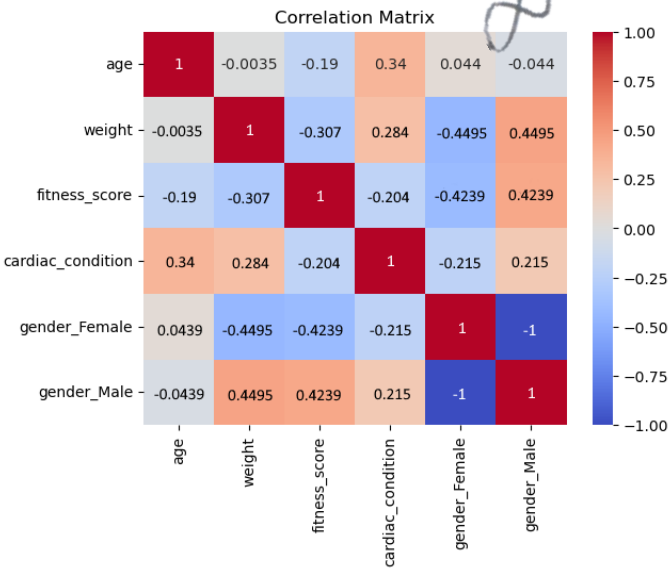


Fig. 15: Correlation Matrix

is used with the help of a correlation matrix table which provides numerical data. None of the correlation values are about 0.5 which suggests that there is hardly any existent correlation among them.

3) Violin plot:

A violin plot is a data visualization that is used to depict the distribution and summary statistics of a dataset, providing insights into its probability density. The violin plot for the features age, weight, and fitness_score is plotted against the dependent feature cardiac_condition. The plot in Fig.16 shows the distribution of the independent variables against the dependent variable. The width of the distribution in the plot indicates the frequency of values at different levels.

4) Kernel Density Estimation plot:

The Kernel Density Estimation plots for numerical features in a data set based on the target variable cardiac_condition are plotted as can be seen in Fig.17.

The KDE plot shows two distributions, one for each binary class of the target variable cardiac_condition which is 0 and 1. The KDE plot for Class 0 shows the distribution of the features when the value of the target variable is 0, and similarly, the KDE plot for Class 1 shows the distribution when the target variable is 1. The KDE plots provide a visual depiction of the numerical features that vary for different classes of the target variable, and analyze the feature distributions for the outcome variable cardiac_condition.

5) Boxplot and Outliers:

Box plot is applied to the data set, only on the numeric data features for identifying outliers and the spread of the data. Outliers are data points that deviate from the rest of the data points present in a data set. Effectively managing outliers is a crucial stage in data processing, given their potential to significantly influence statistical analyses.

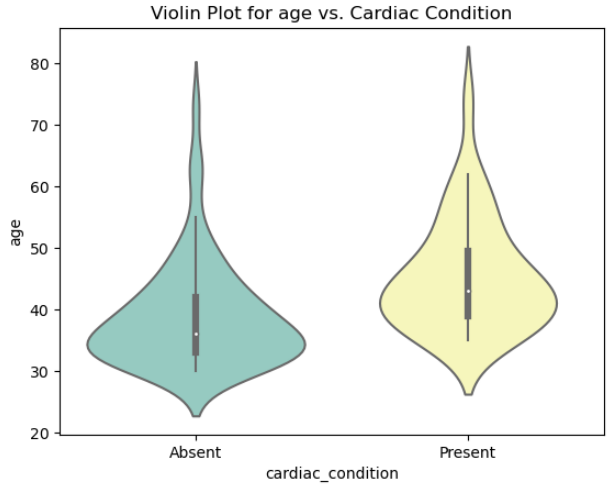
From the boxplot in Fig.18, it can be noticed that outliers are present in the data set only in the Age column. This is because there is some significant amount of skewness present in the data, and it should be removed during data pre-processing. The outlier percentage also shows that there is 4 percent of outliers present and the rest of the features are free from outliers.

Log transformation is used as a technique to reduce the impact of outliers, particularly fixing the right-skewed distributions. Log transformation compresses the scale, making extreme values less influential. After the implementation of the Log transformation, the outlier is checked using the boxplot in Fig.19. The boxplot shows a very drastic change with the majority of outliers removed by using the transformation. Also, the outlier percentage which is checked using the interquartile range, is now reduced to 2 percent, which is far better than the previous observation.

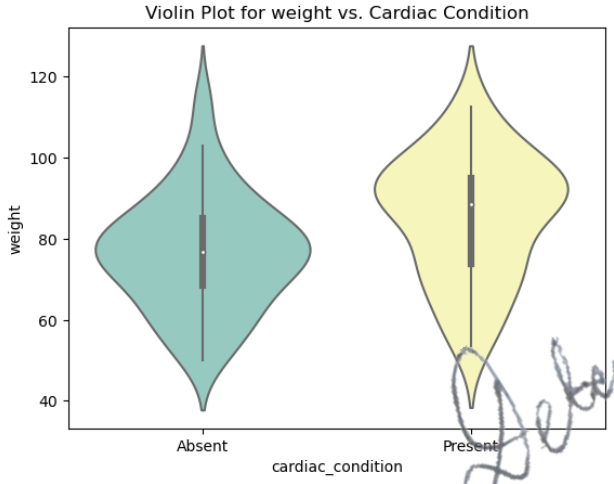
C. Modeling

1) Splitting the dataset:

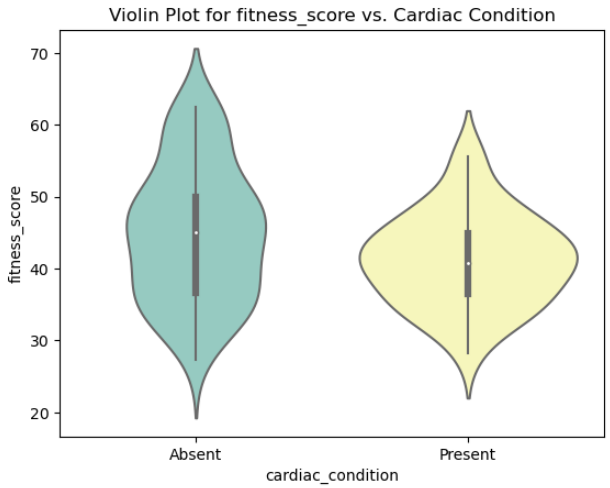
The cleaned dataset is split into two individual datasets, one consisting of all the independent features and the one consisting of the dependent feature 'cardiac_condition'. These



(a) Violin plot for Age



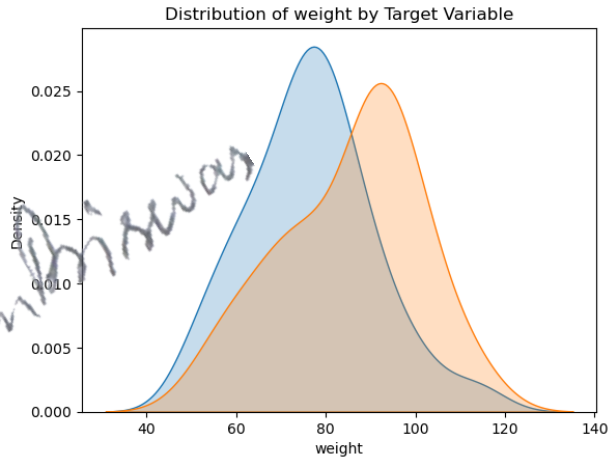
(b) Violin plot for Weight



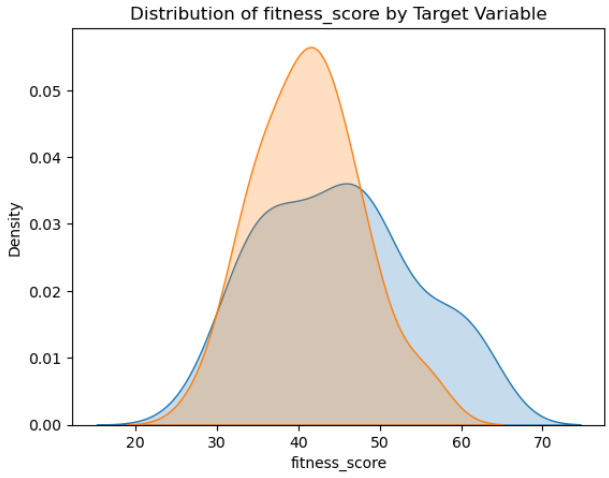
(c) Violin plot for Fitness Score

Fig. 16: Violin Plot

(a) KDE plot for Age



(b) KDE plot for Weight



(c) KDE plot for Fitness Score

Fig. 17: Kernel Density Estimation Plot

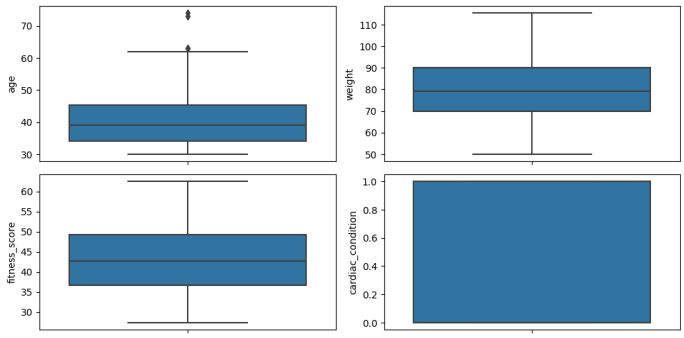


Fig. 18: Boxplot before

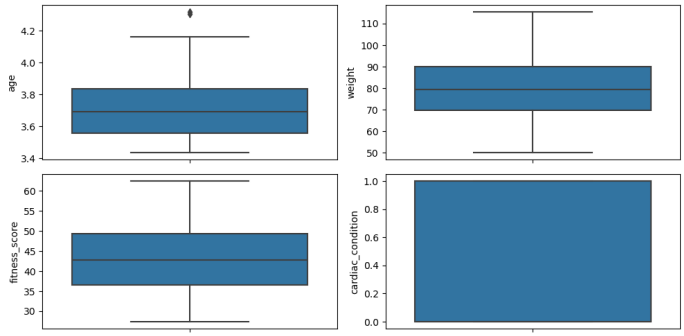


Fig. 19: Boxplot after

two datasets are now split into train and test datasets, based on seed equivalent to the student ID 22242821. The dataset train will fit into the machine learning model and the test data will evaluate the performance of the fitted model. The test size for the split is set at 20 percent for evaluation in the later section.

2) Scaling the dataset:

The StandardScaler is employed to standardize features. Standardization involves transforming the dataset's features to have a mean of zero and a standard deviation of one. This process ensures uniform scaling across all features, preventing any particular feature from disproportionately influencing the learning process solely due to its larger magnitudes. The standardized data will be applied to the Final model in the next section.

3) Model Training:

The train data is now fitted into the Logistic Regression model for further evaluation and analysis. Intermediate and Final model is generated.

a) Intermediate Model:

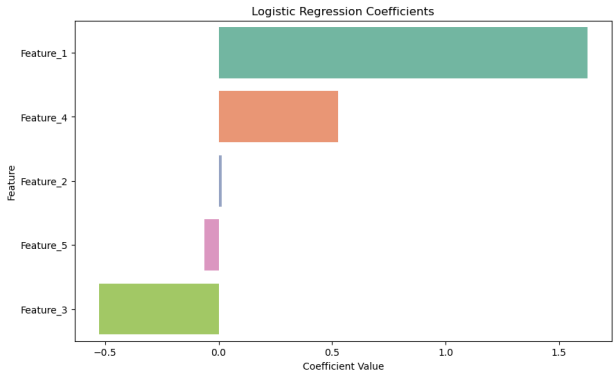
The train data is fitted into the intermediate model for further evaluation and analysis. The intermediate model is created for evaluation. The intermediate model is used as a benchmark for comparison, providing valuable guidance in the ongoing refinement process, and aids in generating the final model.

b) Final Model:

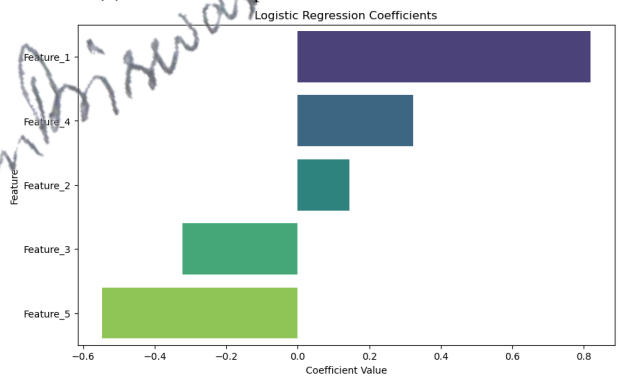
The scaled train data is now fitted into the final Logistic Regression model for further analysis. This model signifies the meticulous optimization and fine-tuning of the hyperparameters. This model stands as the outcome of the development process and proficiently makes predictions.

D. Interpretation

From both the intermediate and final Logistic Regression Model coefficient graph in Fig.20(a) and Fig.20(b) respectively, it can be observed that the coefficients and the Intercepts fetch different values respectively. The value of the Final model provided a better result. The process of model inter-



(a) Co-efficient plot for Intermediate Model



(b) Co-efficient plot for Final Model

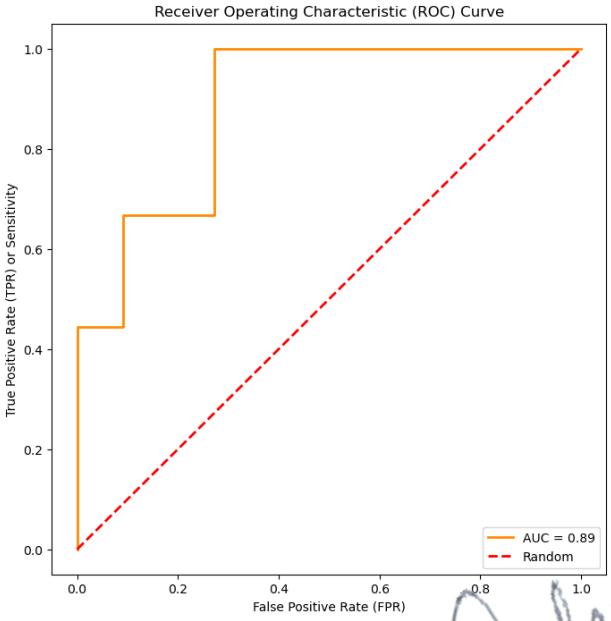
Fig. 20: Co-efficient Plot

pretation, and understanding the coefficients and intercept is essential for understanding the logistic regression model. The coefficients of the intermediate model show a greater range of values than the final model. Larger magnitude coefficients indicate a stronger influence of the corresponding feature. The intercept in the intermediate model is much smaller than the one in the final model. This suggests there's been a noticeable change in the starting prediction for the positive outcome.

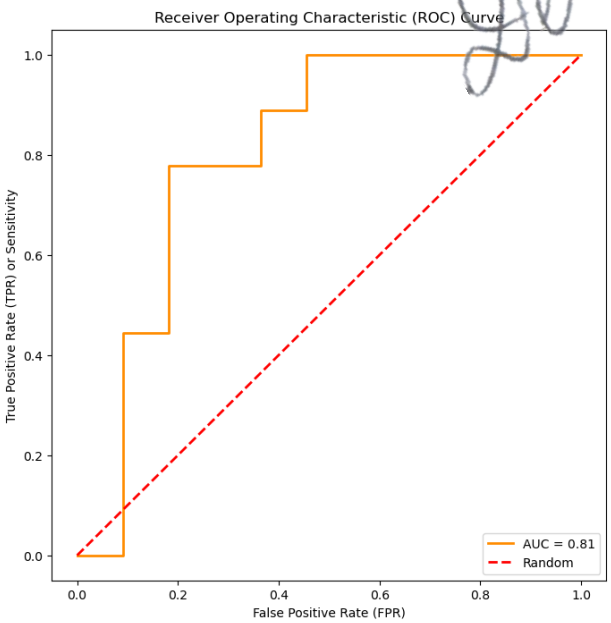
E. Diagnostics

The ROC (Receiver Operating Characteristic) curve is a visual representation that illustrates the diagnostic and performance characteristics of a binary classification. The ROC represents the true positive rate on the vertical y-axis, representing the proportion of actual positive instances correctly identified by the model, and False Positive Rate indicates the

proportion of actual negative instances incorrectly classified as positive. The Area under the curve is a measure of the overall performance of the ROC curve. An AUC value close to 1.0 indicates superior model discrimination. The AUC value for both models is close to 1, as can be seen in Fig.21(a) and Fig.21(b), which indicates a better model performance in determining the positive class and negative class. The final evaluation of the model's performance will depend on the classification reports based on Accuracy, Precision, Recall, and F1 score.



(a) ROC for Intermediate Model



(b) ROC for Final Model

Fig. 21: Receiver Operating Characteristic Curve

F. Evaluation

The Logistic regression model gives two slightly different results for the Intermediate and Final Models. The evaluation of the dataset using logistic regression is dependent on some of the classification metrics.

1) Accuracy:

It is the proportion of correctly classifying the instances. The Accuracy for the intermediate models stands at 0.75 whereas the final model is 0.8 suggesting that the final model provides better overall correctness of predictions.

2) Precision:

It represents the ratio of correctly predicted positive instances to all instances predicted as positive. Fig.22(a) and Fig.22(b) illustrate that the Final model exhibits greater prediction values than the intermediate model for both classes. This indicates a lower occurrence of false positives in the final model.

```
Accuracy: 0.75

Classification Report:
              precision    recall  f1-score   support

           0       0.71      0.91      0.80        11
           1       0.83      0.56      0.67         9

    accuracy                           0.75        20
   macro avg       0.77      0.73      0.73        20
weighted avg       0.77      0.75      0.74        20
```

(a) Evaluation of Intermediate Model

```
Accuracy: 0.8

Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.91      0.83        11
           1       0.86      0.67      0.75         9

    accuracy                           0.80        20
   macro avg       0.81      0.79      0.79        20
weighted avg       0.81      0.80      0.80        20
```

(b) Evaluation of Final Model

Fig. 22: Model evaluation

3) Recall(Sensitivity):

It denotes the ratio of correct positive predictions to the total number of actual positive instances. The final and intermediate model has almost the same values except for class 1 holds a bit higher value suggesting that the final model has a somewhat better identification of actual positives.

4) F1 Score:

It represents an equilibrium between precision and recall. A higher F1 score reflects a superior balance and a reduced likelihood of class imbalance, as observed in the final model. The higher F1 score of the final model determines the final model to perform better in terms of prediction accuracy.

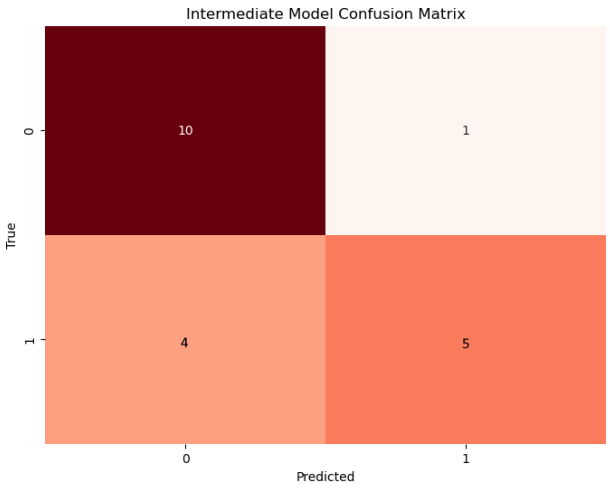
5) Area Under the Receiver Operating Characteristic Curve (AUC-ROC):

It quantifies the model's capacity to differentiate between the

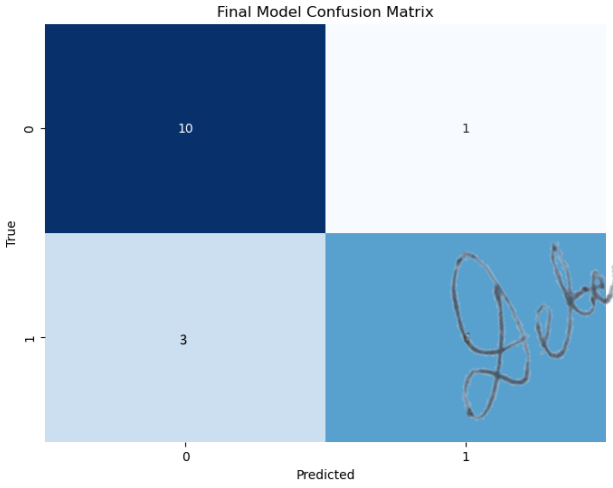
two classes. Both the models have higher values of AUC close to 1 indicating better discrimination between classes.

6) Confusion Matrix:

A confusion matrix is a table used in classification to evaluate



(a) Confusion Matrix for Intermediate Model



(b) Confusion Matrix for Final Model

Fig. 23: Confusion Matrix

the performance of ML models. It provides a summary of a model's predictions in a classification scenario, displaying the counts of true positives, true negatives, false positives, and false negatives. The confusion matrix in Fig.23(b) for the Final model shows a higher value of false positive and false negative than the intermediate model depicted in Fig.23(a) thus stating the final model's performance.

IV. Final Conclusion Of Dataset A and Dataset B

A. Time Series Analysis

A time series analysis strategy was applied to the Weather data set to create a proper forecasting model for future prediction of weather. Simple Moving Average, Exponential Smoothing, ARIMA, and SARIMA were used as time-series models. However, there was some autocorrelation and non-normality present in the dataset. The results of all the models were evaluated and out of all, Exponential Smoothing provided a better result in terms of Forecasting and root mean squared error (RMSE) which is fairly low at 3.03 as compared to SARIMA which is at 3.495. The mean squared error(MSE) and mean absolute error(MAE) are on the lower side for the Exponential Smoothing model which indicates better prediction accuracy. ARIMA did not provide a proper forecast as the data set used was seasonal and hence its evaluation was not taken into consideration.

B. Logistic Regression Analysis

A binary logistic regression strategy was used on the Cardiac data set to assess independent variables that can impact and influence the dependent variable 'cardiac_condition'. Both the intermediate and final models were implemented to fetch the best outcome of the model. The final model showed better performance which was assessed using the necessary parameters of Logistic regression. The final model showed an accuracy of 0.8, a precision average of 0.81 for both the classes, a recall average of 0.80, and an F1 score average of 0.80. AUC was also on the higher side for the final model as the values in the Confusion matrix. Thus confirming that the final model is accurate and efficient for analysis. The dimensional reduction was also implemented before evaluation but it provided a poor result having an accuracy of 0.7 which is fairly lower than the intermediate model, hence was not taken into consideration.

Acknowledgments

The completion of this project has been made possible through the support of Professor John Kelly, for his guidance, advice, and critical feedback.

References

[1] *Applied Statistics*. Deterministic and forecast-adaptive time-dependent models.
by G. Box, Abraham, B, Oxford University Press

[2] Time Series Analysis, by James D. Hamilton, edition 1, 1994, Princeton University Press ISBN: 0691042896

[3] Forecasting: principles and practice, by Rob Hyndman and George Athanasopoulos, Edition 2

[4] Outlier Analysis, by Charu C. Aggarwal, Edition 2, Springer Publications

[5] Logistic Regression- A Primer, by Fred C. Pampel, Edition 2, 2020, Sage Publications

[6] Geeksforgeeks *Logistic Regression in Machine Learning*.
https://www.geeksforgeeks.org/understanding-logistic-regression

[7] Plotly *Violin Plots in Python*.
https://plotly.com/python/violin/

[8] ROC Curves for Continuous Data, by Wojtek J. Krzanowski, David J. Hand, edition 2, 2017, Chapman and Hall/CRC ISBN: 978-1032477732

[9] Python for Data Analysis, by W. McKinney, edition 2, 2017, O'Reilly Media publishers

[10] Analytics Vidhya website *One Hot Encoding vs Label Encoding by Alakh Sethi*.
http://tiny.cc/av_encoding/

[11] Github, Python Data Science Handbook, by Jake. VanderPlas,
https://jakevdp.github.io/PythonDataScienceHandbook/

[12] Medium, A Deep Dive into Seaborn's Kernel Density Estimation Plots: Visualize Data Distributions, by Dr Nilimesh Halder,
http://tiny.cc/kde_seaborn