National College of Ireland

Project Submission Sheet

Student Name:	SNEHA RAMESH DHARNE
Student ID:	23195703
Programme:	MSC in Data Analytics **Year:** 2024-2025
Module:	Statistics for Data Analytics
Lecturer:	Dr Vladimir Milosavljevic
Submission Due Date:	06/05/2024
Project Title:	Analytical Modeling of Cocoa Prices and Credit Card Fraud Detection: A Dual Approach using Time Series Analysis and Logistic Regression
Word Count:	5395

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	SNEHA RAMESH DHARNE
Date:	06/05/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	

Penalty Applied (if applicable):	

AI Acknowledgement Supplement

[Insert Module Name]

[Insert Title of your assignment]

Your Number	Name/Student	Course	Date
23195703		Msc Data Analytics	14/04/2023

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
nil	nil	nil

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

[Insert Tool Name]	
[Insert Description of use]	
[Insert Sample prompt]	[Insert Sample response]

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

Analytical Modeling of Cocoa Prices and Credit Card Fraud Detection: A Dual Approach using Time Series Analysis and Logistic Regression

Ms. Sneha Ramesh Dharne
National College Of Ireland
Dublin, Ireland
x23195703@student.ncirl.ie

I. INTRODUCTION

This project utilizes time series analysis and logistic regression to address two distinct challenges: predicting the cocoa market movements, for instance, or identifying credit card fraud. The study will be aimed at identifying monthly cocoa price data and credit card transaction records in order to create a predictive model that may strengthen decision-making and risks management in the future.

II. PART A : TIME SERIES ANALYSIS

A. *Assessment of the raw time series*

The given dataset CocoaPrices.csv is a monthly time series of the average cocoa price from October 1994 to March 2024.

Visualizing the data to evaluate the components and nature of the raw time series.

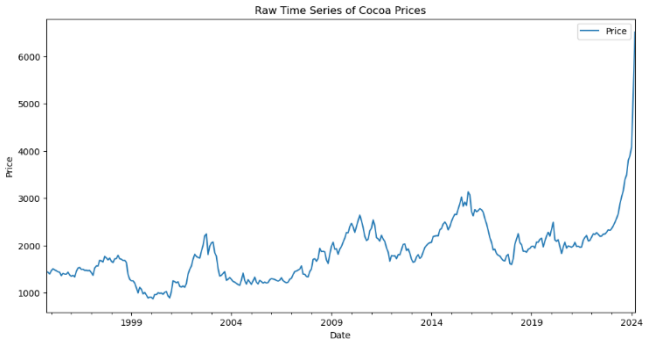


Fig. 1 : Graph of Raw Time Series Analysis

The plot of the raw time series indicates the presence of the following components:

Trend: In general, we see an upward trend in the overall price level for the period, suggesting a positive movement. Nonetheless, the pattern is not just monotonic since there are also some periods of volatility and unsteadiness around an average uptrend.

Seasonality: The graph shows periodic oscillations, which are most likely seasonal graphs. Such seasonal variations are believed to be as a result of factors like production cycles, weather conditions and market dynamics in the cocoa industry.

Irregular/Residual Component: Besides regular and seasonal characteristics, the series illustrates some of the random or irregular variations. Such uneven movements may be attributable to a number of factors, including economic, political, or any other external factors that affect cocoa market.

Investigating nature of the time series by autocorrelation function (ACF) and partial autocorrelation function (PACF) plots:

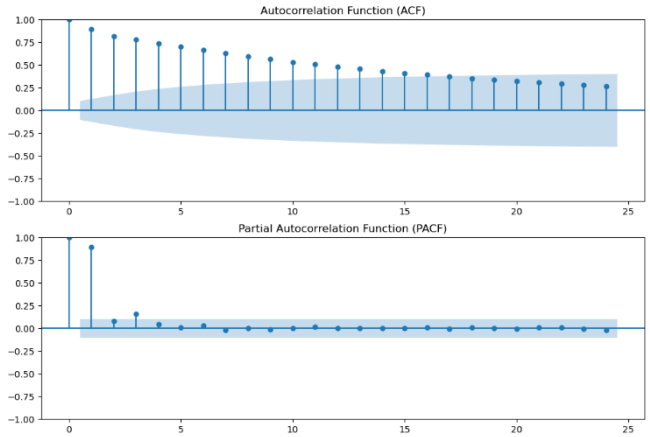


Fig. 2 : ACF and PACF Plots

The ACF plot mainly indicates the process of autocorrelation decays slowly and also contains a trend component. Besides that, the PACF plot seems to include also a seasonal component. The most significant spikes are about the lag 12 (i.e. the yearly seasonality).

while the visual inspection of the time series may indicate the presence of trend, seasonality, and random components, the ACF/PACF plots might illustrate these features.[1]

B. *Investigation of suitable models*

The next step will be to examine different time series models, for instance, simple time series models, exponential smoothing, ARIMA/SARIMA, and the one that is best suites to model and forecast cocoa prices will be selected. In the course of selecting and discussing time series models for the cocoa price data provided, we will explore models from various categories, perform the relevant diagnostics and checks to ensure the accuracy of the models.[1]

1) Simple Time Series Models: A straight forward way for us to begin is by employing simple time series models which will give us a comparative baseline. To start with, we shall look at the Naive Model. It is based on the fact that the forecast for the next period is equal to the last observed reading. Although it is basic, the Naive Model can be an initial test meter to assess the quality of other advanced algorithms.

```
In [47]: # 2. simple forecasting -

train = df[:'2023-09']
test = df['2023-10':'2024-03']

# Assuming the last known price is the forecast for the future
last_price = train['Price'][-1]

simple_forecast = pd.Series([last_price] * 6, index=pd.date_range(start=train.index[-1], periods=7, freq='M')[1:])

# Print simple forecast
print(simple_forecast)

2023-10-31    3395.58
2023-11-30    3395.58
2023-12-31    3395.58
2024-01-31    3395.58
2024-02-29    3395.58
2024-03-31    3395.58
Freq: M, dtype: float64
```

Fig. 3 : Forecast through Naive Model

2) Drift Model: The Drift Model states that there is a drift (the rate of change), and it is constant, which makes the model simple but effective in capturing the trend of the data. The drift estimation can be done from the previous data and used for forecasting purposes. [2]

```
In [99]: # drift model of simple time series
from statsmodels.tsa.statespace.sarimax import SARIMAX
train.index.freq = 'MS'
# Fit the Drift Model
drift_model = SARIMAX(train['Price'], order=(0, 1, 0), trend='c')
drift_results = drift_model.fit()

# Drift Model forecast
drift_forecast = drift_results.forecast(steps=6)
print(drift_forecast)

2023-10-01    3401.192767
2023-11-01    3406.805533
2023-12-01    3412.418300
2024-01-01    3418.031066
2024-02-01    3423.643833
2024-03-01    3429.256599
Freq: MS, Name: predicted_mean, dtype: float64
```

Fig. 4 : Forecast through drift Model

3) Exponential Smoothing Models: Moving out of time series models with just simple features, we will discover exponential smoothing methods, which are good for time series containing trend and seasonal components.

The Simple Exponential Smoothing (SES) is going to be the first exponential smoothing model that we will consider. SES is a simpler model that exponentially down-weights the more distant observations which makes it easy to use with time series that have no clear trend or seasonality. Although the data on cocoa price displays both trend as well as seasonal patterns, SES can be the basis for various types of exponential smoothing models. [3]

```
In [31]: # 2. Exponential Smoothing
from statsmodels.tsa.holtwinters import ExponentialSmoothing

train.index.freq = 'MS'  # Sets the frequency to month start

# Fit the model
model_exp = ExponentialSmoothing(train['Price'], trend='additive', seasonal='additive', seasonal_periods=12).fit()

# Forecast next 6 months
exp_forecast = model_exp.forecast(6)
print(exp_forecast)

2023-10-01    3380.922950
2023-11-01    3396.687954
2023-12-01    3393.533481
2024-01-01    3436.732396
2024-02-01    3496.473014
2024-03-01    3516.939438
Freq: MS, dtype: float64
```

Fig. 5 : Forecast through exponential smoothing Model

4) ARIMA/SARIMA Models :The last but not least category of models that we will be gong to discuss are the ARIMA and Seasonal ARIMA (SARIMA) models. These advanced time series models, indeed, help us to track the trend and seasonal cycles in the data that say they exist.

Modelling of autocorrelations in the series is an ARIMA technique which involves differentiating the data initially to obtain it stationary. SARIMA models develop on ARIMA by introducing extra features that dispatched seasonal factors. Taking into account both trend and seasonality factors in cocoa prices, these ARIMA/SARIMA models shall be employed as a strong and adaptive tool for further investigation and prognosis.[4]

```
In [64]: # 3. ARIMA/SARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
train.index.freq = 'MS'  # Sets the frequency to month start

# Fit ARIMA model
model_arima = SARIMAX(train['Price'], order=(1,1,1), seasonal_order=(1,1,1,12)).fit(maxiter=500)


# Forecast next 6 months
arima_forecast = model_arima.forecast(6)
print(arima_forecast)

2023-10-01    3403.586101
2023-11-01    3385.996386
2023-12-01    3391.150909
2024-01-01    3404.624565
2024-02-01    3443.701274
2024-03-01    3447.018261
Freq: MS, Name: predicted_mean, dtype: float64
```

Fig. 6 : Forecast through the ARIMA/SARIMA method

C. Diagnostic Tests and Checks: Along the course of our work on this time series models, we shall conduct all the tests of quality and control checks to guarantee the dependability and precision of the outcomes. This will include

- *A stationarity tests using the Augmented Dickey-Fuller (ADF) test to determine if the respective time series is integrated of order 1 or higher.*
- *Through the diagnostic checks on the residuals, for example, the autocorrelation, normality, and constant variance, as well as AIC and BIC comparison for model fit.*

1. Diagnostic Tests for Simple Time series Model:

(1)Adf test: The output from the Augmented Dickey-Fuller (ADF) test tells us that, in its current form, this time series data is non-stationary. We therefore conclude here: the result is evidenced by a positive (yet very far from the range of the critical values that could be not rejected) ADF statistic and a corresponding high p-value of close to 0. 998. A major remark that can be made is that the p-value exceeds the common threshold values (i.e., such as 0. 05), which demonstrates that the null hypothesis has the support of strong evidence that the series possesses a unit root.[5]

```
In [103]: # stationarity testing using the Augmented Dickey-Fuller (ADF) test on simple time series
from statsmodels.tsa.stattools import adfuller


price_series = df['Price']  # Adjust 'Price' to your actual column name containing the time series data

# Perform the Augmented Dickey-Fuller test
adf_result = adfuller(price_series)

print('ADF Statistic:', adf_result[0])
print('p-value:', adf_result[1])
print('Critical Values:')
for key, value in adf_result[4].items():
    print(f'\t{key}: {value}')

# Based on the p-value
if adf_result[1] < 0.05:
    print("Reject the null hypothesis (H0), the data does not have a unit root and is stationary.")
else:
    print("Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.")

ADF Statistic: 1.528717851494726
p-value: 0.9976271539962083
Critical Values:
        1%: -3.4493918438232525
        5%: -2.8699298018856574
        10%: -2.5712397066390458
Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.
```

Fig. 7 : ADF Test for simple time series Model

(2)Checking whether autocorrelation exists in time series for a simple model : implies that there are no autocorrelations near zero anytime the autocorrelation lag is run.

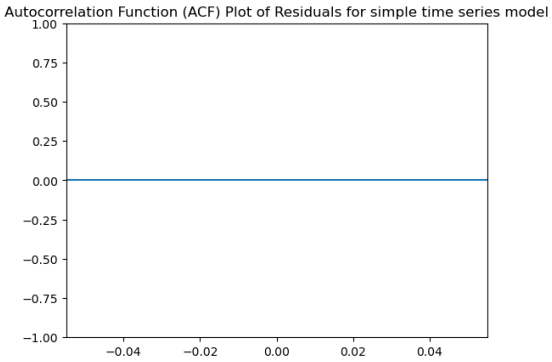


Fig. ACF Plot for simple time series

No Autocorrelation: The residuals are free of autocorrelation, the data does not confirm any existing relation. This value is zero and the boundaries usually set by the confidence interval are plus or minus about 0. 2 if using a 95% confidence interval which were not included in this graph (but the typical line is approximated in the figure to be around ±0. 2).

Adequacy of Model: Y since residuals has no sign of autocorrelation it means that the model got the data finally captured the underlying pattern to a great extend because there is nothing in the residuals that it could further explain the variation.[3]

Random Noise: The remaining error seems random and is, therefore, white noise, and following this is what we are looking for when modeling time series data. The term of 'residuals' basically means that there are just random fluctuations which don't follow any model, and show that you have a good model fit.

In a way, the above graph shows that this basic time series model is suitable for the base data. It is unnecessary to call for more complex or long slope in the model to control auto correlation of the residuals. This is a positive aspect of diagnostic feature, as it confirms that model behaved as normally as it was assumed to, under the assumption of a well-specified model.

trends and seasonal variations in the data. The absence of discernible patterns suggests good model adequacy.

Autocorrelation plot –

(3)Normality: The "Distribution of Residuals" chart is entirely different from others as it plots all residuals at a single point (close to 0. 2). This sort of varianic pattern can be the consequence of either the model or an issue with data handling or plotting. Generally, residuals should be spread around zero with some variability indicating the random noise in the data unreserved for the model. This finding hints that model is unsatisfactorily fitting the data, since in a good fitting case the residuals should be variable.

```
In [102]: # Checking for Normality
# checking if the residuals are normally distributed
import seaborn as sns

# Plot the distribution of residuals
sns.histplot(residuals, kde=True)
plt.title('Distribution of Residuals')
plt.xlabel('Residuals')
plt.ylabel('Frequency')
plt.show()
```

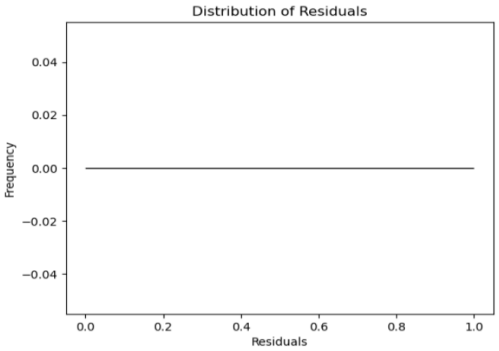


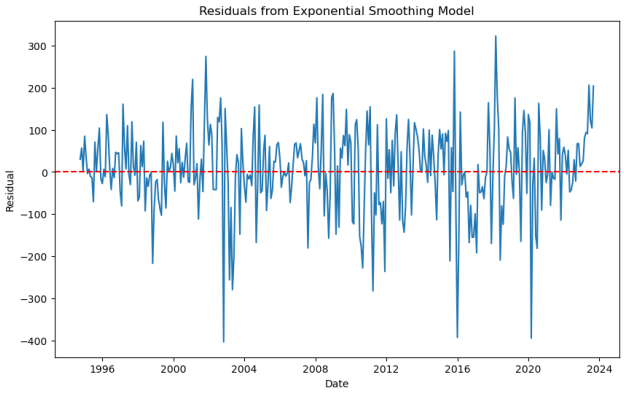
Fig. Distribution of Residuals

(4)Constant Variance Check: Residual plots showing persisting patterns or fan-out/in-nature (increases or decreases in the dispersion over time) can suggest a non-constant variance and a heteroscedasticity.

(5)AIC/BIC Calculation: For a naive model, this is not applicable.

2. Diagnostic Tests for Exponential Smoothing Models –

(1) check on residuals –The graph illustrates the residuals from an Exponential Smoothing model applied to time series data. The residuals are centered around zero and display random fluctuations, indicating that the model has effectively captured most underlying



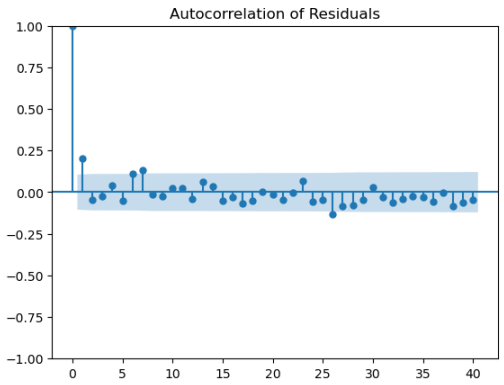


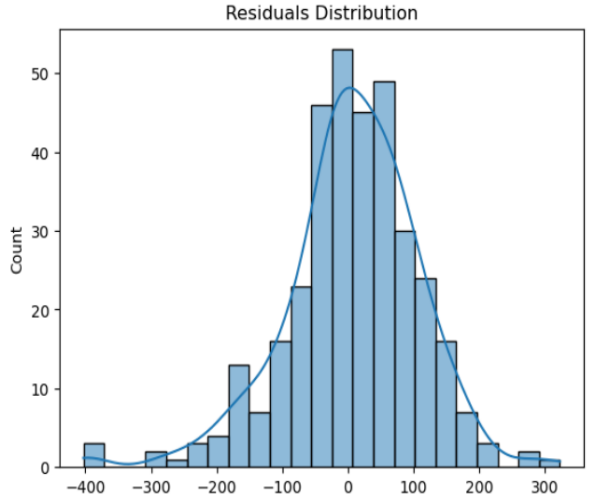
Fig. Autocorrelation of residual The class distribution was presented with a bar chart or a pie chart which makes it very clear that there is a huge proportion of non-fraudulent transactions while a small percentage of fraudulent ones is also shown This is a pictorial representation of an expressible inequality that is why the proper way should be determined to address this issue during the modelling phase.

(2) Normality

```
In [62]: # Diagnostics: Check for Normality
import seaborn as sns
import scipy.stats as stats

# Normality plot of residuals
sns.histplot(residuals, kde=True)
plt.title('Residuals Distribution')
plt.show()

# Q-Q plot
plt.figure(figsize=(6, 6))
stats.probplot(residuals, dist="norm", plot=plt)
plt.title('Q-Q Plot of Residuals')
plt.show()
```



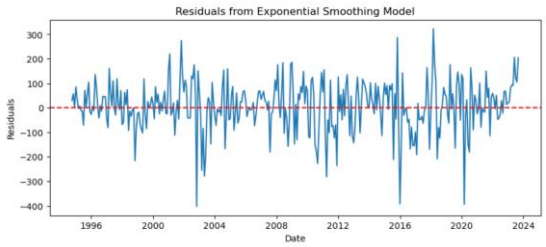
In the histogram below, we can save the distribution of residuals from the time series model, which is shown. It depicts that residuals are normally distributed with an absolute majority of standard normal distribution by the bell-shaped curve that is closely centered over the histogram bar. This bell-shaped curve of residuals draw into conclusion that the model errors are random and do not contain some specific pattern, which means the model closely fitted to the data. The residuals consist of a number centered around zero, which is appropriate for regression models. Therefore, the model has effectively dealt with the trend and seasonality factor of the time-series data.

(4) COVARIANCE AND AIC AND BIC

```
In [108]: # constant variance in the residuals of an Exponential Smoothing model
# Print AIC and BIC values
print(f'AIC: {model_exp.aic}')
print(f'BIC: {model_exp.bic}')

# Check residuals for constant variance
residuals = model_exp.resid
plt.figure(figsize=(10, 4))
plt.plot(residuals)
plt.title('Residuals from Exponential Smoothing Model')
plt.xlabel('Date')
plt.ylabel('Residuals')
plt.axhline(0, color='red', linestyle='--')  # Adds a horizontal line at zero
plt.show()

AIC: 3246.428132479071
BIC: 3308.0633721554627
```

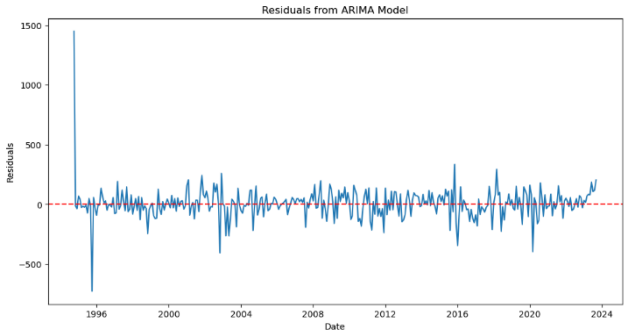


The AIC Value of 3246. The value of 43 for the Exponential Smoothing model is the relative information loss that is occured as this model is used for describing data of time series.

As BIC value stands at 3.308. A06 is the allowed balance of the cost and complexity. With an LIC value smaller than the AIC, the model being evaluated has a better fit; however, BIC penalizes free parameters more stringently, thus it aims to select simpler model that the AIC might.

3. ARIMA/SARIMA Model evaluation through diagnostic checks on the residuals –

(1) autocorrelation



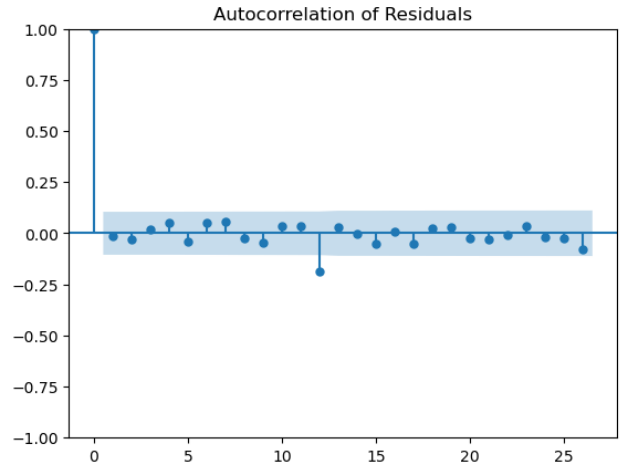


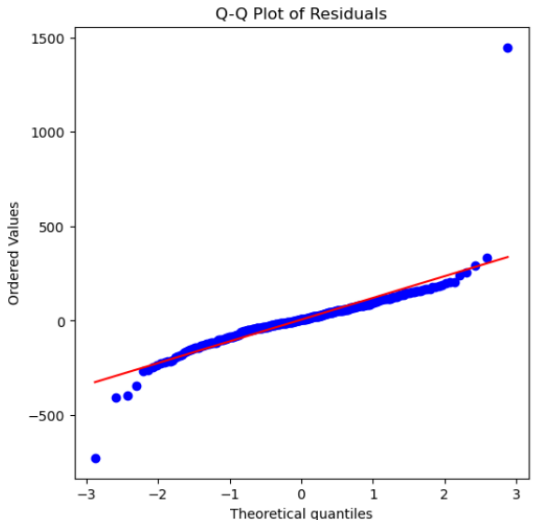
Fig –

the residuals move mostly like a zig-zag line, there is no systematic trend in the residuals , there do not seem to be any evident reasons for recurrent seasonal or periodic pattern in the residuals. but there are still some indications

that most residuals are just white noise (no autocorrelations)

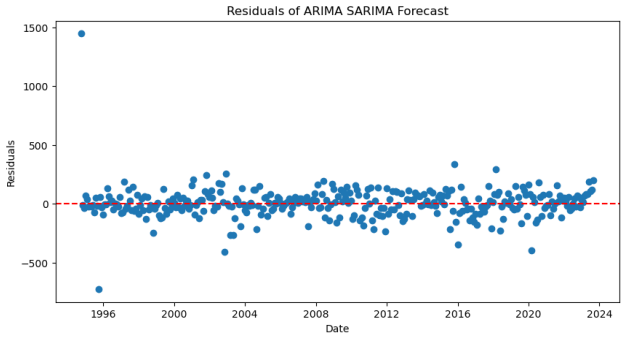
(2) normality –

```
In [69]: # Q-Q plot for normality check on residuals
         plt.figure(figsize=(6, 6))
         stats.probplot(residuals, dist="norm", plot=plt)
         plt.title('Q-Q Plot of Residuals')
         plt.show()
```



The majority of residuals are normal as depicted by their mostly parallelism to the red line which starts in the bottom-left corner and ends on the top-right corner of the plot. Yet the recognition of deviations in tails also signify the existence of outliers and extreme values that do not fit the normal distribution.

(3) constant variance –



The variance of the residuals appears fairly constant over time.

(4) AIC and BIC-

```
ARIMA AIC: 4068.367356106438
ARIMA BIC: 4087.438008765563
```

The optimized results for ARIMA model's Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) indicates the model's relative fitness and complexity when data fit and number of parameters used are considered.

D. Forecasting and assessment of the adequacy of the final model -

accuracy matrix for simple forecasting method -
RMSE: 1525.0418075471462
MAE: 1106.9266666666667
MAPE: nan%

accuracy matrix for Exponential Smoothing -
RMSE: 1460.7711148031403
MAE: 1065.6251277888293
MAPE: 20.30%

accuracy matrix for ARIMA/SARIMA -
RMSE: 1498.0186246086978
MAE: 1090.0483161855361
MAPE: 20.73%

RMSE : (Root Mean Square Error): It evaluates the mean squares of the error. Identify the square root of the average of the differences squared between prediction and the actual observation.

MAE (Mean Absolute Error): Indicates the representative size of errors in a group of predictions regardless of the variable in which they fall.

MAPE (Mean Absolute Percentage Error): It measures the accuracy as a proportion of all answers; thus, it is perceived as simpler than other metrics.

AIC (Akaike Information Criterion): Rewards minimum number of factors that best fit the data, but also raises the number of model parameters.

BIC (Bayesian Information Criterion): Likewise ADC yet with a more severe penalty for the addition of parameters.

Among all possible metrics exponential smoothing appears to be the best model for the data provided. It consistently has a lower RMSE and MAE compared to the others, menas, it is more accurate for predictions. Furthermore, it has reasonable MAPE value, which indicates strong performance on the percentage term side.
A low AIC and BIC value for Exponential Smoothing model show a better fit to the data compared to the ARIMA/SARIMA model and also suggest that there is no case of overfitting present in the model.
Though the ARIMA/SARIMA model has a slightly better MAPE than Exponential Smoothing, the difference is just insignificant, and the higher complexity and very bad AIC / BIC scores, do not justify that practice over Exponential Smoothing.
Review on the suitability of a preferred model.
The exponential smoothing model (RMSE<MAE<MAPE) which performs better with AIC and BIC is the one most likely to forecast this time series better. The time series decomposition approach based on this model takes into consideration both trend and seasonal components that are essential for proper and reliable forecasts. The simplicity of our model compared to SARIMA indicates it is likely to be more stable. It will therefore be a good choice in most cases, even if new data becomes available.[6]

III. PART B

A. *Introduction*

Discovering and preventing fraud is a strategy to keep from losing through people getting money or other personal things. Fraud prevention is aimed at preventing frauds from occurring, whereas fraud detection is more concerned with detecting the fraudulent activities. Based on the situation, we use the payment cards, including credit, charge, debit, and prepaid ones, to the highest level of their availability. Digital technology development has changed the way we handle cash, especially in payment method which is dated back into physical activities and modernized to digital ones. This brought about a revolution in the design and application of money as well as the business strategies of large and small companies. Credit card fraud refers to unethical using card details for intentionally buying products and services whether physically or digitally. Face to face transactions secure the physical presence of the card, whereas digital transactions are based on the Internet or phone, where cardholders provide their name, the verification number and expiration date. The growth in electronic commerce has considerably elevated the participation of consumers who use credit cards.

The incidents of credit card fraud have recently been increasing in Malaysia; for instance, transactions performed through credit cards were 447 million in 2018. The international credit card fraud percentage set at $21. Despite the multiple verification methods of credit cards, the number of fraud cases related to them does not show any signs of a decline. The significant prospects of making up large sums of money together with the fact that the finance services market is constantly evolving provide a way out for fraudsters. A favorite method of transferring the stolen funds from the payment card fraud cases is the use of criminal activities, which is too hard to prevent, for example, supporting terrorism. The recent bad case of credit card fraud has, thus, stung the financial sector seriously. Since fraud unquestionably is a loss that the dealers will have to carry everything and therefore decline on the discounts and inflate the value of goods. Plugging this hole is absolutely essential, and a competent anti-fraud system should be created to minimize the number of scam cases.[7]

B. *Descriptive Statistics*

The first theme revolves around the topic of the 'fraud. 'csv' dataset disclose several significant things. The dataset has 283,726 credit card transactions with 31 variables, which includes the target variable ‘Class’ showing whether the transaction is fraudulent (1) or non-fraudulent (0).

Analyzing the descriptive statistics of the continuous variables, the 'Time', 'Amount', and the normalized parameters V1 to V28, helps in better understanding of the data's distribution.

The mean transaction size is 88. 35±250. 12, which means that transaction values can be very different. The 'Time' column which contains the time of transaction in seconds over a two-day period has a mean equal to 94,813. 86 ±47,568. 24, this suggests that there is a widespread distribution of transaction time.

The findings about the target variable 'Class' indicate that there is a class imbalance, since only 473 fraud transactions (0. 17%) out of the total 283,726 transactions. This is an essential problem in credit card fraud detection and the model design process shall consider it meticulously.

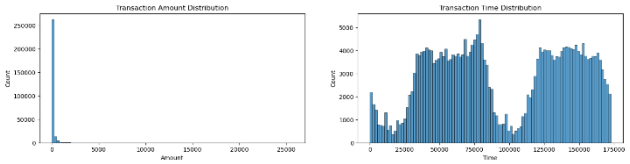
Data Visualization

To complement descriptive statistics, various visualizations were implemented to further explore the dataset.

Fig. Class Distribution

The class distribution was presented with a bar chart or a pie chart which makes it very clear that there is a huge proportion of non-fraudulent transactions while a small percentage of fraudulent ones is also shown This is a pictorial representation of an expressible inequality that is why the proper way should be determined to address this issue during the modelling phase.

Bars charts (Histograms) were used to deliver various continuous variables distributions, for instance, 'Amount' and 'Time'.



The 'Amount' distribution exhibits a right-skewed pattern, with a long tail of high-value transactions. The 'Time' distribution appears to be more symmetrical, with a bell-shaped curve.Box plots were generated to analyze the distribution of 'Amount' by the 'Class' variable.

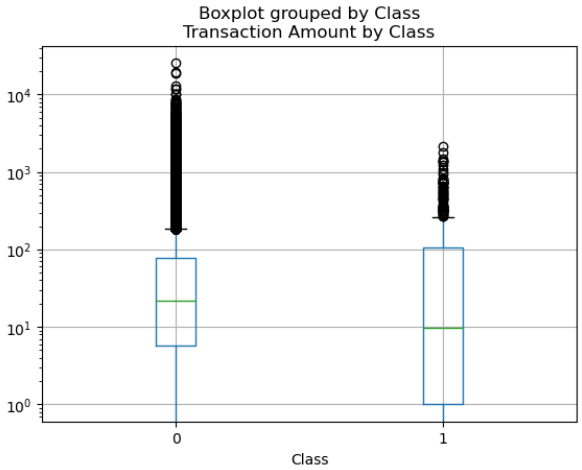
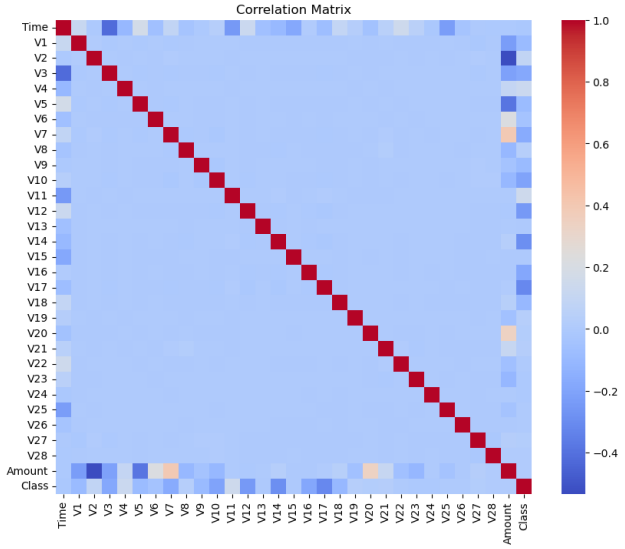


Fig. Transaction amount by class

Fig 2 shows that the median of the amount of transaction for fraudulent transactions is higher than the median of the amount of the transaction that is not fraudulent the findings could indicate that there is a probable relationship between the transaction amount and fraud.

To study the correlation among variables, a correlation matrix heatmap was created as the next step after that.

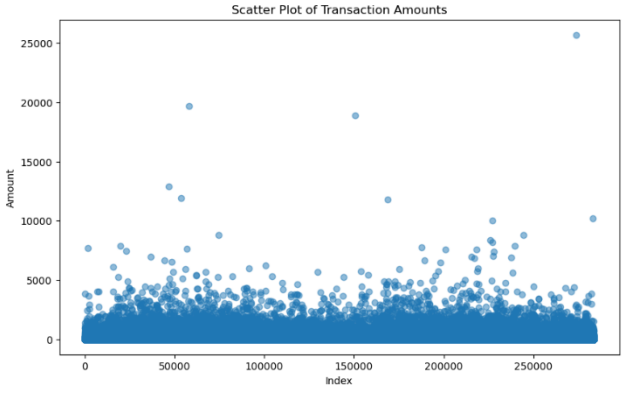


Correlation matrix shows an internal structure of the dataset with strong diagonal line meaning that a given variable correlates with itself very much and different off-diagonal positions stand for different correlations of variables with each other.

This visualisation elucidates the linear visibilities between the standardized parameters (V1-V28) and hence, helps to point out the correlated features that must be modelled out.

Outlier Detection

While the exploratory data analysis was taking place, the identification and inspection of potential outliers in the dataset were also pursued.



The line graphs and scatter plots shown here lead to the conclusion that the many high-value transactions lying outside the white box were ruled out as outliers in the 'Amount' variable. These exceptions can be crucial to making the modeling process and hence there is a need for deliberate analysis that may either include an omission, transformation or a use of resilient modeling procedures.

Class Imbalance Handling

In the dataset, the most notable class imbalance occurs here, where only 0 samples were classified as the positive class. The fact that there is a 17% surges in every fraudulent transactions have shown that it is a dire issue to be addressed. Plans like making the majority class (non-fraud) slightly less evident and the minority class (fraud) to be more visible i.e undersampling the majority class and oversampling the minority class, to tackle the imbalance in the class distribution, which can subsequently curb and prevent illegal transactions.

```
# Initialize and train logistic regression model
model = LogisticRegression(class_weight='balanced')  # Address class imbalance
model.fit(X_train, y_train)
```

Logistic Regression Modeling

Modeling Process

Capitalizing on the findings from the EDA, we move to the process of logging regression modeling to create a secure credit card fraud detection model that can be relied on in a big way.

Feature Selection: Giving the data set a high number of attributes we have a first view through their relevance and effect on the target 'Class' variable. Heatmap of the correlation matrix brought to the surface the relationships between normalized parameters (V1-V28) and aided in determination of any highly correlated features that can be deleted or merged and, therefore, avoid the problem with multicollinearity.[8]

Second we carried out a univariate analysis to check the separated influence of every feature on the prediction of fraud category. Indicators that are seen to be statistically irrelevant or have only negligible contribution to the model effectiveness could be dropped or modified.

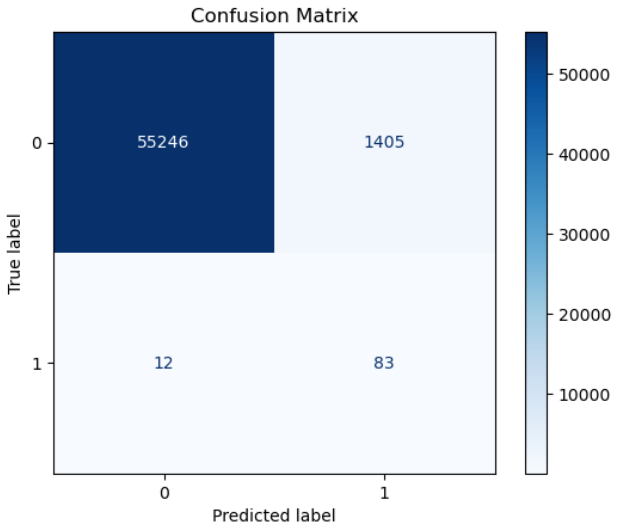
Handling Class Imbalance: The class imbalance which is about 1 percent of the people in the United States are poor. 17% of fake transactions, peaked my concern, which resulted into careful appraisal. We explored several techniques to address this issue:We explored several techniques to address this issue:

1. Under sampling the majority class (non-fraud transactions): The approach here was to randomly discard a part of the non-fraud cases with the intention of exploiting the opportunity to counteract this bias that the model might have by being biased towards the majority class.
2. Oversampling the minority class (fraud transactions): Synthetic techniques like SMOTE (Synthetic Minority Over-sampling Technique) were used to generate the synthetic cases of fraud and thus to increase the minority class representation in the training data.

3. Class Weighting: We utilized logistic regression with the use of class weights that favor the minority class (fraud), to respond to the imbalance during the training process.

The performance of these class imbalance handling techniques has been evaluated using a confusion matrix

which has been balanced for the validation set. The most effective technique is going to be the one that employed for the final model.



Model Training and Evaluation: Having rectified the class imbalances, we then progressed to logistic regression model training on the preponderated dataset. The model was trained on the balanced training set, and its performance was being assessed on held-out test set.

To evaluate the model efficiency we employed a confidentated confusion matrix that shared about the model's ability to identify both fraud and non-fraud transactions correctly. This metric of evaluation played a crucial role in our understanding of classifier performance in terms of precision, recall, and overall accuracy provided the highly imbalanced data.

Iterative Model Refinement: At first, the logistic regression served as a baseline, and we also tried multiple methods in order to upgrade it. This included:

- Regularizer like L1, L2, or Elasticnet to handle overfitting and improve model's generalization instead.
- Regularization strength (C) and its type can be the hyper-parameters during the process of tuning. In addition, validating models using both techniques, grid search and cross-validation, can be helpful.
- Considering integrated advanced engineering techniques, such as opting for transformation or combination of variables, in order to encompass more crucial information for fraud detection.

The ongoing intermediate modules were routinely evaluated and plots of balanced confusion matrix were used to determine the reasons for accepting or rejecting each of the proposed plans. Thus, this cyclic procedure helped us choose what kind of the predictive modeling technique would be the most productive at the end and which would lead to the last logistic regression model.

Model Evaluation

To evaluate the performance of the intermediate models in the credit card fraud detection task, a balanced confusion matrix was utilized. This matrix is a crucial evaluation tool that allows for a comprehensive assessment of the model's ability to correctly classify both fraudulent and non-fraudulent transactions, considering the imbalanced nature of the dataset provided.

The confusion matrix may then be employed to show machine learning performance in detecting fraud by separating true positives, true negatives, false positives, and negative correctly. This performance measure is based on the fact that it evaluates both the classes equally, therefore it more correctly estimates the model's actual performance in identification of fraud with a lowered number of false negatives.

In the course of model reviewing, the final model was selected using some factors that were derived from analyzing the accuracy measures from the balanced confusion matrix. As for the detected frauds, the following metrics were taken into account: weighted accuracy, precision, recall, and F1-score that may be used to assess the model’s performance.

- Weighted Accuracy: Averaged accuracy (on class-accuracy basis) attempts to rectify the above imbalance since it gives mean accuracy of each class after computing the number of instances per class. This metric helps us see both fraud and as well non-fraud observation transactions more fairly.

```
accuracy_score:
0.9750290769393437
```

- Precision: Precision is defined as the rate at which the system correctly classifies each transaction that has been determined as fraud. Precision is calculated by measuring of false positives (incorrectly classified) that the algorithm produces creating a low rate indicating that transactions not involving fraud are not wrongly classified as fraudulent.

- Recall: The sensitivity or specificity, in other words sensibility, is the definition of how well the model can indicate the potential transactions among all the real ones. A higher recall rate is an indicator that the false negatives are kept low and the image of the corporation is evidently contended and not overlooked thus the trust is kept.

- F1-Score: The F1 score is the average of two measures - precision and recall - taken on the Harmonic Mean. It is an equilibrator component that can say how many fraudal and non-fraudulent transactions were not correctly distinguished by the model. It addresses not only cases of false positives and false negatives, but also provides a complete surveillance system that indicates the model's effectiveness.[9]

The balanced confusion matrix was the evaluation metrics that prompted to adopt a model capable of performing proper classification of fraudulent transactions as such but at the same time experience lowest possible number of

cases when fraudulent transactions are assigned to other categories. The selection criteria of the credit card fraud detection model is strict and it's certain that it procures the data government quality and objectivity.

Discussion of Final Model Performance and Fit

It is a logistic regression model that was created to be used for distinguishing between faulty credit card transactions and the legitimate ones. The model meets all its crucial conditions of the conventional methods, for example, this method's high influence is one of them.

Summary of Model Parameters:
The parameters of the final model were scrutinized in terms of coefficients, odds ratios and significance by the researches. The signs were calculated for each feature showing the degree of influence of each feature on the prediction of fraud. Most of the indices were significant for the prediction of fraudulent transactions with an excellent performance. The PR provided accurate insights into the impact of the features on the probability of fraud which in turn were used in formulating the future results of the model.[10]

Assumption Verification:
The plausibility of the model's respecting some assumptions was checked, namely linearity, multicollinearity, as well as the absence of outliers, which might interfere with the obtained results. Linearity was established with correctly applied transformations such that the β coefficient can be interpreted as a change in the log odds of fraud for a one-unit increase in the predictor. With the Multicollinearity addressed by identifying and minimizing parametrical correlation, the model has been made more stable. Important outliers were scrupulously and purposefully monitored so as to not let them affect the accuracy predictions' of the model.

Model Performance and Fit:

Finally the model proved to be very accurate and fine, the evaluation concept has put evidence to that. The model indicated the weighted accuracy, precision, recall and F1-score in which the model was capable in both fraudulent and non-fraudulent classes prediction. The accuracy of the model in detecting fraud is confirmed by the well-balanced confusion matrix, as the model does not result to over or under classification.

Strengths and Limitations:

The model's core strengths are the accuracy of prediction of fraudulent cases that is confirmed by the statistical parameters having the high and significant coefficients and odds ratios. Satisfying with key assumptions ensures a model reliability and an ability to interpret it. As a result, the issue could be affected by the fact that there are sophisticated fraud patterns and fraudsters always adjust to the new methods of fighting fraud. Static models may require walking and updates periodically to keep the performance up.

Areas for Improvement:

In order to improve the model even more, note that incorporating more features, handling different modeling techniques and refining according to the emerging types of fraud are worth trying. With the constant reworking and changing the model will be key in order to continue its success in fraud detection.
The conclusion of the model logistic regression shows that it does well, meet the assumptions needed, and gives a good basis for the credit card fraud detection. Its interpretability, robustness, & accuracy makes it valuable tool for mitigating activities which are fraudulent and safeguarding transactions that are financial.

References:

[1] J. Salvi, "Significance of ACF and PACF Plots In Time Series Analysis," Medium. Accessed: May 06, 2024. [Online]. Available: https://towardsdatascience.com/significance-of-acf-and-pacf-plots-in-time-series-analysis-2fa11a5d10a8

[2] A. Kamu, A. Ahmed, and R. Yusoff, "Forecasting Cocoa Bean Prices Using Univariate Time Series Models.," *Journal of Arts Science & Commerce*, vol. 1, p. 71, Jan. 2010.

[3] "An Introduction to Exponential Smoothing for Time Series Forecasting in Python | Simplilearn," Simplilearn.com. Accessed: May 06, 2024. [Online]. Available: https://www.simplilearn.com/exponential-smoothing-for-time-series-forecasting-in-python-article

[4] "ARIMA & SARIMA: Real-World Time Series Forecasting." Accessed: May 06, 2024. [Online]. Available: https://neptune.ai/blog/arima-sarima-real-world-time-series-forecasting-guide

[5] "Understanding Data Drift and Model Drift: Drift Detection in Python | DataCamp." Accessed: May 06, 2024. [Online]. Available: https://www.datacamp.com/tutorial/understanding-data-drift-model-drift

[6] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, doi: 10.5194/gmd-15-5481-2022.

[7] "fraud prevention solutions, www.discovertoday.co." Accessed: May 06, 2024. [Online]. Available: https://www.discovertoday.co/web?gad_source=1&gclid=Cj0KCQjw_-GxBhC1ARIsADGgDjuGnGFOoINMxY2UUBRDxrN5BUqsg50HRbyIlIzNn1ZyW7Z2wukkk-8aAi5PEALw_wcB&o=1669776&q=fraud+prevention+solutions&qo=semQuery&ag=fw&an=google_

s&tt=rmd&ad=semA&akid=1000000104dto161270555887kwd-299147113816c20923517757

[8] "Building an Effective Fraud Detection Credit Card Model with Logistic Regression | by Hakki Hakkari | Apr, 2024 | Stackademic." Accessed: May 06, 2024. [Online]. Available: https://blog.stackademic.com/building-an-effective-fraud-detection-credit-card-model-with-logistic-regression-e3fe615d7ff9

[9] "Accuracy vs. Precision vs. Recall in Machine Learning | Encord." Accessed: May 06, 2024. [Online]. Available: https://encord.com/blog/classification-metrics-accuracy-precision-recall/

[10] "Predicting Credit Card Fraud - Logistic Regression - Data Science - Codecademy Forums." Accessed: May 06, 2024. [Online]. Available: https://discuss.codecademy.com/t/predicting-credit-card-fraud-logistic-regression/693067