Statistics TABA: Time Series & Logistic Regression

Aniket Kutty Shetty
MSc. Data Analytics (2024-2025)
National College of Ireland
Dublin, Ireland
x23177861@student.ncirl.ie

Abstract: This Paper consists of a deep analysis of two datasets which is executed in python and is divided into two parts. The first part of the paper is about how the study was done in depth on cocoa prices and a time series model was built. The second section of the paper consists of a deep analysis on a credit card transaction to forecast whether the transaction is genuine or fraudulent. A logistic model was built to detect frauds. Many models were compared, and we have chosen the best-fit model for both analyses.

Keywords: Logistic Regression, Time series, Credit Card Fraud Detection, Cocoa Prices

1. Introduction:

Cocoa has a very huge market in the beverages industry and many companies like Starbucks and Barista rely on cocoa supply. The aroma, freshness, quality and price of the cocoa beans has a direct effect on the business of these big players as well. Thanks to our talented farmers who work hard to make sure the cocoa beans are extracted and exported for the world and for all the coffee lovers. In this research I have done lthe analysis of the price variations of cocoa beans data which is from 1995 October to 2024 March using time series model. We have done the Naïve model; we have performed smoothing using various types of smoothing. The Arima and Sarima models have been built to determine which model is the best for this dataset. Root mean square was taken into account for the evaluation of the different models.
The use of credit cards has become popular in recent times. As there are many benefits of using Credit cards, people tend to use credit cards more and more to get various offers. As the demand for credit cards has increased so is the risk of using credit cards.
The age group between 60-80 years young often have trouble using credit cards and these people are the ones who can easily get themselves into some credit card scams.
In the second part of the analysis, we will build a Logistic Model and forecast whether a Transaction of the credit card is fraud or not fraud. We have also compared and evaluated which model is the best one suited.

2. Time Series

a) Description of Dataset:

The Cocoa beans dataset is taken from the online source i.e. International Cocoa Organization which is a monthly time series data from 1995 to March 2024. The data has 354 rows and 2 columns.
"Date" – Consists of the different Dates of the Cocoa price.
"Price" – Average Price of the cocoa beans.

b) Data Analysis:

The First step of this analysis included checking if the data has any null, missing or duplicate values i.e data cleaning then data preprocessing.
There are no missing values present in the dataset.

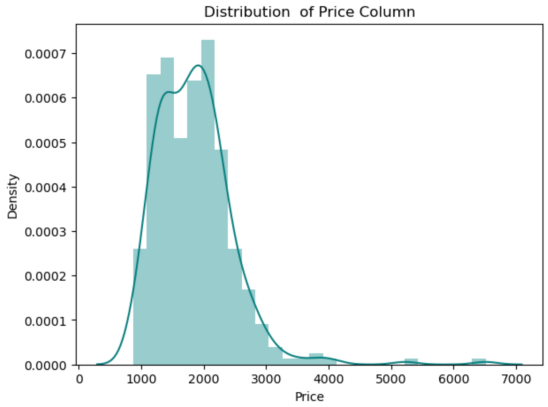


Fig 1.1 Distribution Chart of Price Column

The distribution in the Fig 1.1 of the price column appears to be a normal distribution. Curve.

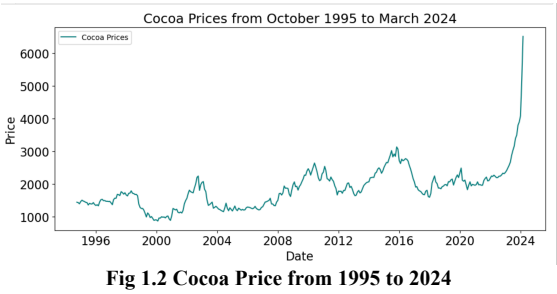


Fig 1.2 Cocoa Price from 1995 to 2024

Fig 1.2 here represents the cocoa price from 1995 to 2024. The graph above shows the double peaks but given the conditions, we're okay with it.
The price vs date shows a clear positive trend. It is possible to see three successive peaks that occur at periodic intervals.

As we know that the components of Time series analysis are

- Trend - means how is data's long-term behavior.
- Seasonality - Seasonal variations in the weather cause short-term repeating patterns in the data.
- Cyclic Variations -Recurring patterns in the same year
- Irregularities – Data variations that are erratic and unexpected.[1]

To evaluate a Time Series and create a prediction we must take into account the aforementioned elements and ensure that our data is unaffected by each and every one of them.
From the Fig 1.2 we can observe that it shows a multiplicative model since the peaks' width and height are changing with time. But still, we will observe closely whether the model is Multiplicative or addictive.

c) Decomposition

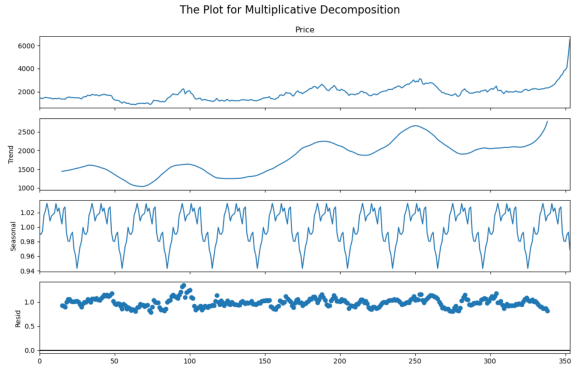


Fig 1.3 Multiplicative Decomposition

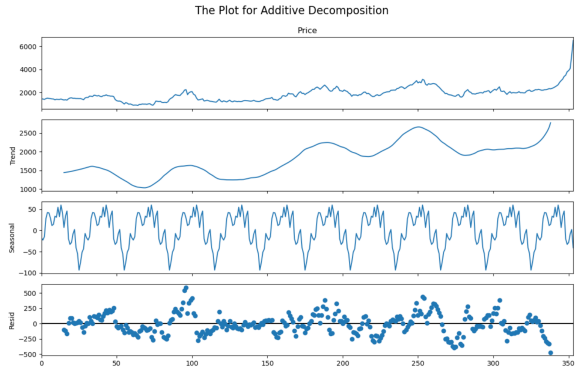


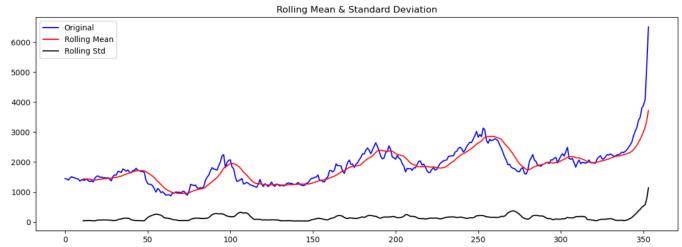
Fig 1.4 Addictive Decomposition

If we compare Fig 1.3 & and Fig 1.4, we can clearly say that there is some remaining or left out patterns in the additive decomposition residuals when we closely examine them.
The multiplicative decomposition seems to be a random, which is the best thing. Therefore, the choice of multiplicative decomposition is best for the study of this data. The linear increase in the data is seen clearly and it is that the seasonal data is present here.

d) Testing the Stationarity in the Data

We have to determine whether or not the dataset is stationary throughout the TSA model preparation phase. Statistical tests are used to do this.
We have used **Dicky Fuller Test** to analyse the stationarity in the data. [1]

Null Hypothesis (H0): the sequence exhibits non-stationarity.
Alternate Hypothesis: the sequence exhibits stationarity.
If p-value greater than 0.05 then we have ignored to reject the null hypothesis.
If p-value is less than equal to 0.05 then we can reject the null hypothesis and accept alternate hypothesis.




```
Results of Dickey-Fuller Test:
Test Statistic                   1.528718
p-value                          0.997627
#Lags Used                       7.000000
Number of Observations Used    346.000000
Critical Value (1%)             -3.449392
Critical Value (5%)             -2.869930
Critical Value (10%)            -2.571240
dtype: float64
```

Fig 1.5 Rolling Mean & Standard Deviation of Time series.

It is evident that the Rolling Mean for the Cocoa Price time series data varies over time. The crests and troughs of the rolling standard deviation change with time.
Test Statistic: (1.52) > Critical Value (5%) : (-2.86)
p-value (0.99) > 0.05.
From the above evaluation we can therefore infer that the aforementioned Cocoa Price Time Series is not stationary and therefore the Null Hypothesis cannot be rejected.

e) Differencing

By subtracting the previous value from the next, or differencing, we can erase trend and seasonality and make the time series stable. From here, we'll work on the data's log values.

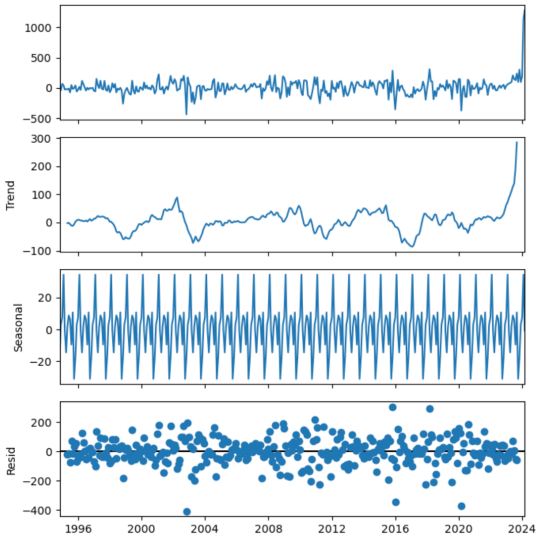
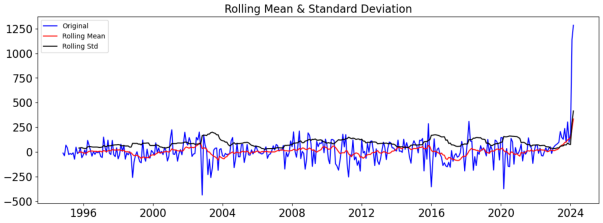


Fig 1.6 Seasonal Decomposition

In order to address stationarity, we have obtained log of the data, and trend and seasonality are addressed using difference. Both the trend and the values of the data have almost completely stopped increasing. But it's evident how seasonal the data is.

We will now verify that the time series(differenced) is stationary by the below plot Fig 1.7




```
Results of Dickey-Fuller Test:
Test Statistic                  0.074481
p-value                         0.964275
#Lags Used                      6.000000
Number of Observations Used   346.000000
Critical Value (1%)            -3.449392
Critical Value (5%)            -2.869930
Critical Value (10%)           -2.571240
dtype: float64
```

Fig 1.7 Rolling Mean & Standard Deviation (Diff)

The Rolling Mean is extremely near to zero based on the results of the Augmented Dickey Fuller Test. Over time, the Rolling Standard Deviation shows a trend of increase and decrease. From the above observations we can see that Critical Value (5%): (-2.86) > Test Statistic: (0.07).
0.05 > p-value (0.96).
We will use the above time series based on the data even if it does not pass the stationarity condition. Double-differencing could result in poor performance and have an excessively negative impact on the data.

f) Model Building for Time Series.

In this study we have built various models, and the data was split into two parts the data from 1995 to September 2023 is used as the training data and the data from October 2023 to March 2024 was taken as test data.

➢ Naïve Model

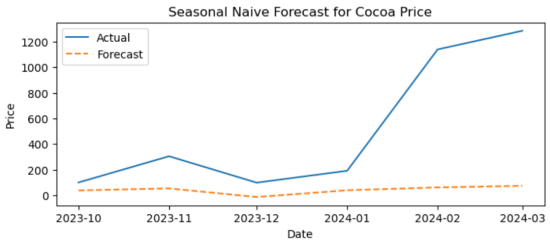


Fig 2.1 Seasonal Naïve Model Forecast for Cocoa Price

We have built a Naïve Model for Cocoa Price forecasting. I have used it because a seasonality can be observed in the data. From the above Fig 2.1 we can say the prediction and the Actual value of price column from October 2023 to March 2024 are not matching. The RMSE value calculated for this model is very high i.e. 647.67.

Exponential Smoothing
Moving Average Method. – A moving average is a technique that uses several averages of various selections to evaluate data points.

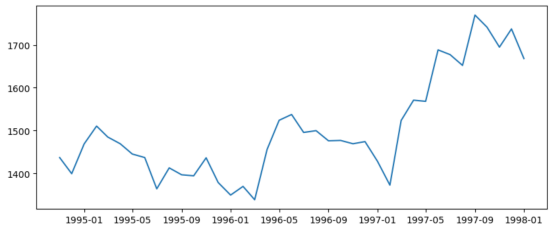


Fig 2.2 Plot from 1995 to 1998 of cocoa price

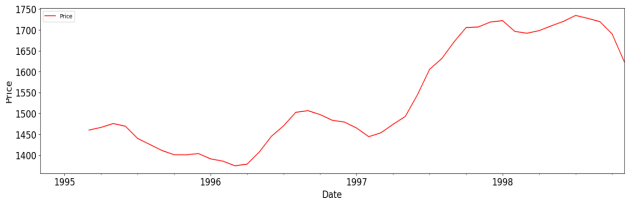


Fig 2.3 Plot from 1995 to 1998 of cocoa price

if you observe the Fig 2.2 and then compare it with & Fig 2.3 the seasonality has been smoothened. The Fig 2.2 is having more variation and red is having less variations.
The Fig 2.2 represents the smoothened time series on moving average and blue is the original time series. We have passed a window as 5 Moving average i.e. going behind in the analyse and taking an average of how many windows we want

➢ Simple Exponential Smoothing.

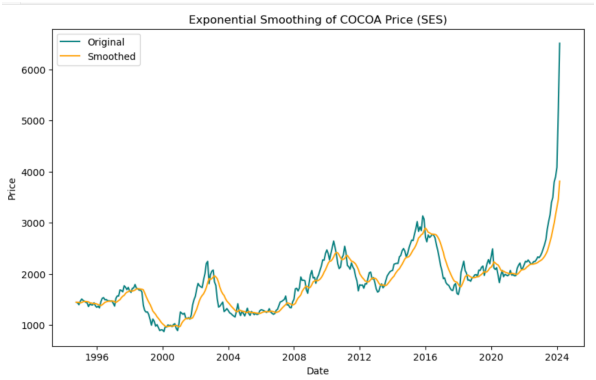


Fig 2.4 Simple Exponential Smoothing.

Here in Fig 2.4 the original cocoa price data is represented by the solid line in Teal colour, while the smoothed data is represented by the orange line. The value of α=0.2 decides how smooth the curve is going to be. The degree of smoothing will increase if the alpha value is increased. The RMSE value for the SES model is 261.

- **Exponential Smoothing by Holt method.**

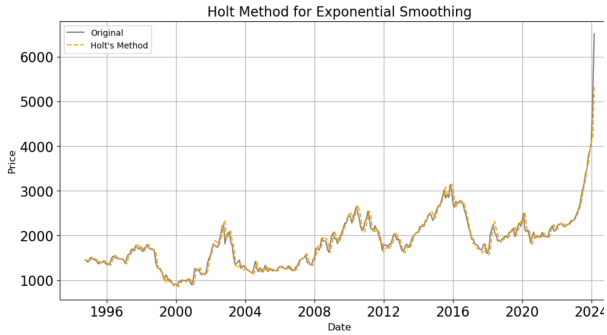


Fig 2.5 Exponential Smoothing by Holt method.

The Fig 2.5 is an Exponential smoothing by Holt's method. By the dashed line in orange colour, we observe the level of the data and data's trend, providing smoother forecasts compared to SES. The Holt method here is suitable only for the data with trend. It is not suitable for the data with seasonality. We have taken alpha value as 0.8 and beta value as 0.2.

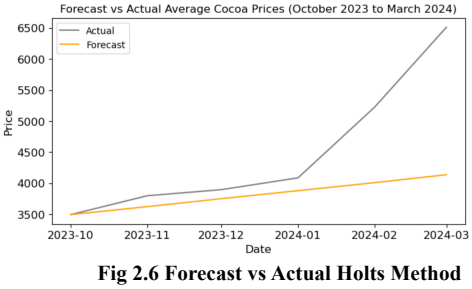


Fig 2.6 Forecast vs Actual Holts Method

Fig 2.6 shows us the actual vs the predicted values by Holt smoothing method. This clearly shows us that the predicted value vs the actual values is nowhere near. The RMSE value for Holt method is Quite high i.e. 1095.

- **Exponential Smoothing by Holt Winter's method**

We have used Exponential Smoothing by Holt Winter's method as our data has seasonality present in it.

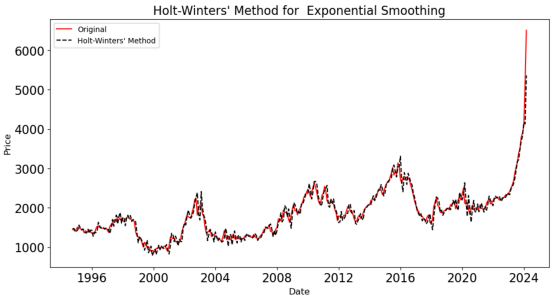


Fig 2.7 Holt Winter's Plot for Smoothing

In Fig 2.7 The Holt-Winters' method line which is in black colour and is represented by dashed line and tells us about the different level of data, different trends in data points and data's seasonality. It gives us the smooth forecasts of data and is usually used for the data with seasonality. We have the following parameters for our model i.e α alpha, β beta, and γ gamma. The Alpha value considered is α = 0.8, beta is β = 0.2, and gamma is γ = 0.6.

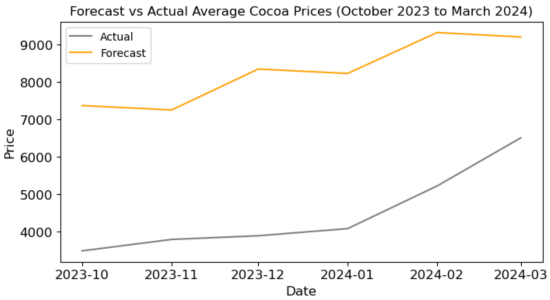


Fig 2.8 Forecast VS Actual Holts Method

The RMSE value before dividing the data into training and testing is 153 and the RMSE value for the Forecasted Holts Winter method is very high and is also higher than holt's method i.e. 3829.

- **ARIMA MODEL**

Arima Model represented as

AR- Autoregressive. (p)

Utilises the dependent connection of a current information and its historical values.

I – Integrated (d)

Differencing our observations.

MA -Moving Average (q)

Utilises the connection between an observation and the residual errors obtained by applying a moving average model to logged observations.

Before using the Arima Model we have to make sure that our data passes the assumptions of the Arima Model.[2]

a) Stationarity – Over an extended length of time, the mean and variance of the normally distributed series remain constant.
b) Uncorrelated random error – The variance and mean of the data remain constant over time, with the error term being randomly distributed.
c) No outliers – As outliers can affect the conclusion strongly and can be confusing.
d) Random Shocks - Shocks are regarded as randomly distributed whenever they occur, with a constant variance and a zero mean.

In the prediction equation, p denotes the number of lag observation which we can find by PACF plot, d the number of times the observations are differenced & q is the order of moving average found by ACF plot.

- **Auto-Correlation Function (ACF)**

The correlation between a data point and the lag values is summarized in the visual. The y-axis displays the correlation value for positive (+) and negative (-) correlation, within -1 and 1, while the x-axis displays the lag values.

- **Partial Auto-Correlation Function (PACF)**

The Plot summarizes the correlation for observations with the lag values that the previous lag observation does not explain .

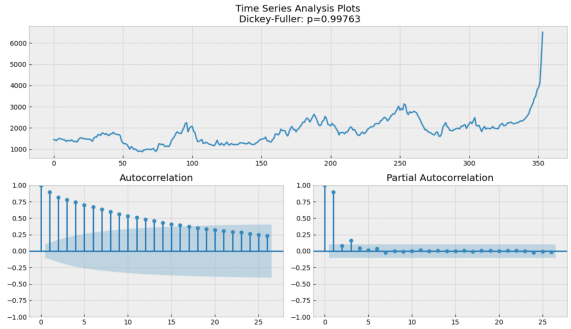


Fig 2.9 Plot of ACF & PACF for original data

The Fig 2.9 depicts the correlation between the time series and its lags. A positive decreasing correlation is observed in this case.
The first lag, which is likely the most important lag, is seen to be outside of the confidence interval in the PACF graph above. It most likely determines the ACF graph's pattern, with each subsequent lag following the preceding lag.

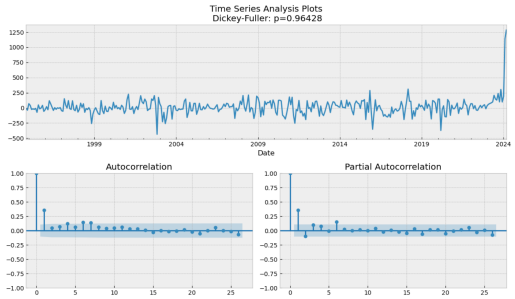


Fig 2.10 ACF & PACF after Differencing

From the above Fig 2.10 I have selected the value of p, q and d for ARIMA model.

The degree of differencing done in our case is 1. The PACF graph helps us to find the value of p i.e. 2 in our case as 2 values are in the significant area. The ACF help us find the value of q i.e. 2.

```
ARIMA MODEL Results Predicted VS Actual
predicted = 3443.673223, expected = 3495.030000
predicted = 3488.127481, expected = 3799.150000
predicted = 3852.408061, expected = 3897.040000
predicted = 3906.725435, expected = 4087.540000
predicted = 4157.024289, expected = 5226.120000
predicted = 5574.676643, expected = 6510.160000
```

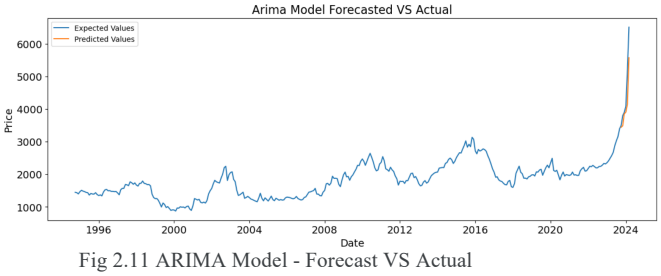


Fig 2.11 ARIMA Model - Forecast VS Actual

```
1 RMSEValue = np.sqrt(mean_squared_error(test,myArimaPredictions))
2 print('Test RMSE: %.4f' % RMSEValue)
```

```
Test RMSE: 598.9092
```

Fig 2.12 ARIMA Model – RMSE Score.

By Fig 2.11 we can say that our Arima Model has performed well. The forecasted values is closer to the actual values which is a good sign for any time series models. The RMSE Value for our ARIMA model compared to other models has given a good output i.e. 598.90.

➢ SARIMA Model.

Seasonal Auto Regressive Integrated Mis an expansion of the ARIMA model that can manage the data's seasonal impacts is the SARIMA model. In SARIMA Model we have P, D, Q, M which is the seasonal order where the P D Q are same as that of ARIMA model. M represent the number which want to provide for a one seasonal cycle.[2]

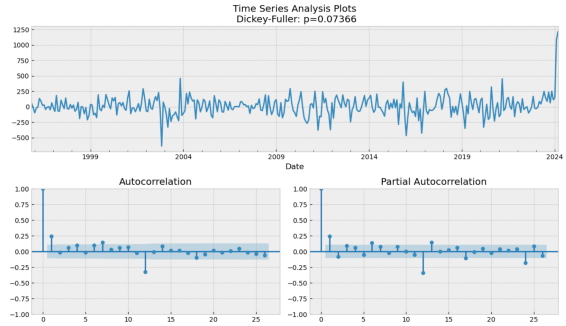


Fig 2.13 ACF and PACF after Double differencing

From the above Fig 2.13 the p value has decreased to 0.07 but still it not less than 0.05.

```
SARIMA MODEL : In - Sample Forecasting

predicted = 3363.408475, expected = 3495.030000
predicted = 3348.673482, expected = 3799.150000
predicted = 3357.263191, expected = 3897.040000
predicted = 3365.157845, expected = 4087.540000
predicted = 3409.653684, expected = 5226.120000
predicted = 3409.243419, expected = 6510.160000
```

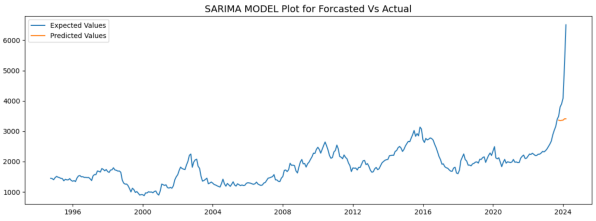


Fig 2.14 Plot for SARIMA Model Actual vs Forecasted.

The Plot of forecasted value clearly tells us that SARIMA is not the best fitted model for our analysis. The RMSE Score for SARIMA model is 1524.72

```
RMSESARIMA = np.sqrt(mean_squared_error(test,sarimapredictions))
print('Test RMSE: %.4f' % RMSESARIMA)
```

```
Test RMSE: 1524.7230
```

Fig 2.15 RMSE for SARIMA Model

3. Logistic Regression

A predictive analytic method called logistic regression is used to classify tasks into binary categories. For example, determining if an email is spam, etc. Before we built the analyse we need to make sure that the assumptions of the logistic model are met.

1st Assumption is the target variable of the model must be of type categorical. For example, in our data we have 2 class either transaction is fraudulent -1or non fraud – 0.

2nd The Data size should be sufficient size before oversampling is done. In our case the data size is 283726 after the pre-processing is done which is enough for us to procced for our analysis.

3rd Linearity of the Independent Variable.

4th Absence of significant outliers: The outliers can influence the performance of our model so we must make sure that there are no outlier present in the data. In our case we have verified and there were no outliers present in the dataset.

a) Description of the Dataset.

This paper has made a use of python to study about the fraud transactions. The credit card transaction data consists of 283726 rows and 31 columns.

Amount: Represents the total transaction amount in Euros.

Class: Whether the transaction was fraud -1 or non-fraud - 0.

Time : The transaction time.

V1.. V8: Normalised Parameters.

b) Data Cleaning

First of all we checked for whether there are any null values or missing values present in our dataset. Then we checked for the outliers in our dataset.

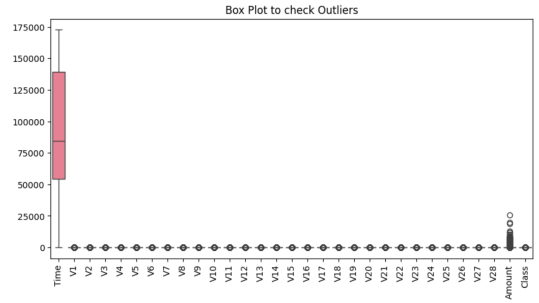


Fig 3.1 Checking for outliers.

As we can observe in our data there is little outliers in the Amount column, but it is okay in this case if we ignore this as it won't impact our results much.

c) Data Analysis & Visualisations

The Skewness of the model can be checked to verify our variables are distributed normally or skewed.

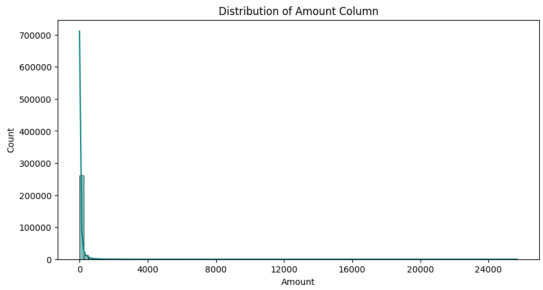


Fig 3.2 Distribution of Amount Column

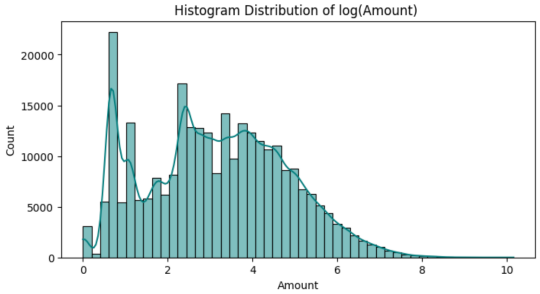


Fig 3.3 Distribution of Log (Amount) Column

The Fig 3.2 says that the Amount column is majorly distributed between points 0 to less than 4000 and very few data are observed in rest of the points from 8000 to 24000. We can say that the distribution is right skewed with the tail stretching in the right. The log of Amount column is taken to get away from the skewness problem which is observed in the Fig 3.2.

From Fig 3.3 we can see the log distribution of the "Amount" column where somewhat the data distribution has somewhat become normal.

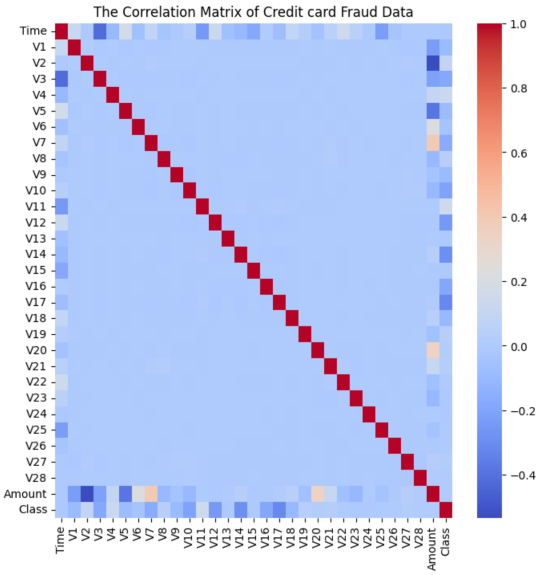


Fig 3.4 Correlation of Variables

The fig 3.4 Shows us the correlation matrix of our data.

From the Correlation matrix I can observe that "Amount" is highly correlated with v1,v2,v3,v5,v6, v7, v20

The target variable "Class" is highly correlated with V2,V4,V17,V14 columns.

Not Fraud %	99.83	Fraud %	0.17
count	283253.00	count	473.00
mean	88.41	mean	123.87
std	250.38	std	260.21
min	0.00	min	0.00
25%	5.67	25%	1.00
50%	22.00	50%	9.82
75%	77.46	75%	105.89
max	25691.16	max	2125.87

Fig 3.5 Percentage Split of total Transactions

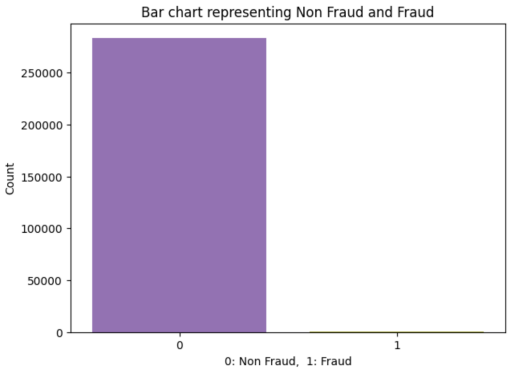


Fig 3.6 Bar Chart for Fraud & non-Fraud

From the Figure 3.5 & 3.6 It can be observed that our dataset of credit card transactions is highly biased. So, to deal

Either there might be possibility that data is missing some information or is inaccurate or it is not capturing the full story. So to deal with such highly biased dataset we can do under sampling of the entire data or oversampling.

d) Model Building.

Logistic Regression: In this study the performance is evaluated on the basis of F1 score of the model and model's accuracy.

- F1 score

A model's accuracy on a binary classification test that takes precision and recall into account.

The formula for F1 score is:

$$2 * \frac{Precision * Recall}{Precision + Recall}$$

The possible values of F1 scores are 0 to 1. A score of 1 means that the model has worked well and can correctly classify each observation. A model with a score of 0 is unable to categorize any data into the relevant class.

- Accuracy

Accuracy refers to how well the model predicts the class. Ratio of cases (true positives and true negatives) properly predicted to all instances.

The accuracy values go from 0 to 1, with 1 denoting good predection and 0 denoting no accurate predictions.

So before performing any kind of under sampling or over sampling techniques I created a logistic model with the original dataset to see how my model is performing. The 70% of the data was given in training variable and the rest 30% was given for test variable.

1st Model performance

```
# Lets run the 1st Model and get classification Report
pred = RunModel(lr, X_train, y_train, X_test, y_test)
```

```
Accuaracy: 0.9992833478230222

Precision: 0.8636363636363636

Recall: 0.608

F1 Score: 0.7136150234741784
```

Fig 4.1 Logistic regression results for original dataset

On the Test data the 1st model gave a accuracy of 0.99 and F1 score is 0.71. The fig 4.2 represents the confusion matrix for our model.

Confusion Matrix

The confusion matrix is a 2*2 matrix which consists of a summary of prediction vs the actual true outcome of the model.
The top left area displays the no. of truly negatives. i.e. values that were correctly predicted as negative by the model.
The top-right area displays the no. of false positives i.e. incorrectly predicted as positive by the model
The bottom-left area displays the no. of false negatives i.e. data which are incorrectly predicted as negative by the model.
The bottom-right area displays the no. of truly positives i.e. correctly predicted as positive by the model.

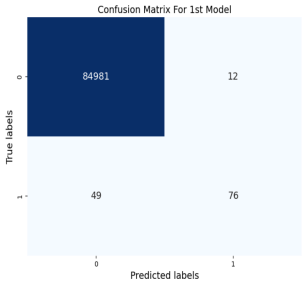


Fig 4.2 Confusion matrix - Original Data

➢ Under Sampling:

Process of taking certain observations out of the class that is majority is called as Under Sampling

```
underSampleData['Class'].value_counts() # Both the class are equal now
```

```
Class
0    473
1    473
Name: count, dtype: int64
```

Fig 4.3 Shape After Under Sampling

2nd Model performance

```
# Lets run the 2nd Model and get classification Report – Model performance measures
print("Under Sample performance measures:")
pred = RunModel(lr, X_train, y_train, X_test, y_test)
```

```
Under Sample performance measures:

Accuaracy: 0.9473684210526315

Precision: 0.9893617021276596

Recall: 0.9117647058823529

F1 Score: 0.9489795918367347
```

Fig 4.4 Logistic regression results for Under sampling dataset

We performed a under sampling for our data and fitted our logistic model. On our test data we received an accuracy of 0.94 and the F1 score is 0.94 as well which is closer to 1. We can consider this model performed well in this case.
Also the according to classification report f1 score for our fraud and non-fraud data is 0.95 which is good.

```
print(classification_report(y_test, pred))
```

	precision	recall	f1-score	support
0	0.91	0.99	0.95	88
1	0.99	0.91	0.95	102
accuracy			0.95	190
macro avg	0.95	0.95	0.95	190
weighted avg	0.95	0.95	0.95	190

Fig 4.5 Performance Measures of Under sampling

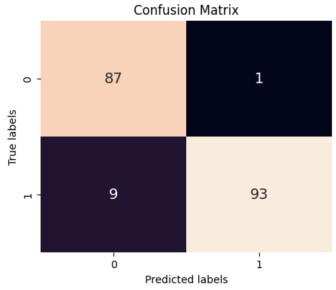


Fig 4.5 Confusion Matrix of Under sampling

From the above confusion matrix, we can say that in the test data the model correctly identified the 93 fraud transaction but missed 9 of them. It also identified 87 non fraud transaction and missed 1.

➢ Over Sampling:

Adding extra instances of the minority class to the data is known as oversampling.

We have used pythons sklearn's package and resample function for oversampling our data

```
# Display class counts
print(OverSampled_DF['Class'].value_counts())
```

```
Class
0    283253
1    283253
Name: count, dtype: int64
```

Fig 4.6 Shape After Over Sampling

Here we divided the 80 percent of data as our training data and 20% of the data as a testing data. When we fitted the model using logistic regression

the accuracy score on test data is 0.94 and F1 score is 0.94 which is same as under sampling.

```
Over Sample performance measures:
 Accuaracy: 0.9441227868881397
 Precision: 0.9675263362523837
 Recall: 0.9194700541910057
 F1 Score: 0.942886268707882
```

Fig 4.7 Performance measures of Over Sample

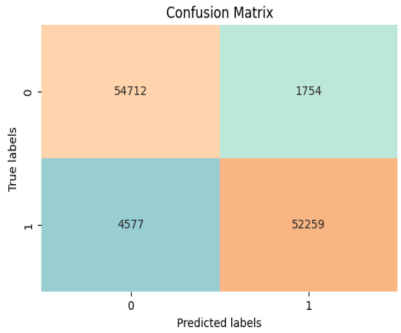


Fig 4.8 Confusion Matrix - Over Sampling

From the above confusion matrix, we can say that In the test data the model correctly identified the 52259 fraud transaction but missed 4577 of them. It also identified 54712 non fraud transaction and missed 1754.

➢ Oversampling using SMOTE

We Also performed Over sampling using SMOTE is Synthetic Minority Oversampling Technique. SMOTE is a technique of oversampling in which samples that are artifical are created for the minority class. This method helps to solve the overfitting problem caused by random oversampling.
It takes the feature space as its focal point and interpolates across positively aligned instances to create new examples.

We performed oversampling using SMOTE technique as well using python's imblearn library. At first, we faced the challenge with this package that corrupted our sklearn library. After several try's I found out the solution and created a separate virtual python environment for my project and ran the codes.

```
SMOTE – Over Sampling performance measures:
 Accuaracy: 0.975110765917636
 Precision: 0.9878266837646974
 Recall: 0.9622442337127677
 F1 Score: 0.9748676541361424
```

Fig 4.9 Performance measures for SMOTE Over sampling

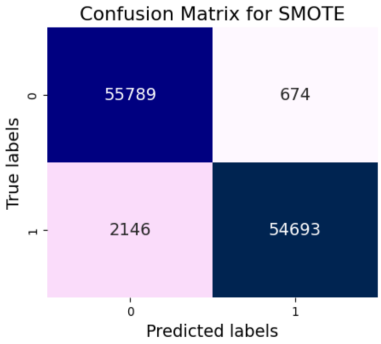


Fig 4.10 Confusion Matrix for SMOTE Over sampling

Using the smote Over sampling method really improved the model performance.
From Fig 4.9 it can be observed that the F1 score for model is 0.97 and accuracy also is same which is better than both the above models.
The confusion matrix in the Fig 4.10 says that model correctly identified the 54693 fraud transaction and missed 2146 points of them. It also identified 55789 non fraud transaction and missed only 674.

```
print(classification_report(y_test, pred))
```

	precision	recall	f1-score	support
0	0.96	0.99	0.98	56463
1	0.99	0.96	0.98	56839
accuracy			0.98	113302
macro avg	0.98	0.98	0.98	113302
weighted avg	0.98	0.98	0.98	113302

Fig 4.11 Summary of SMOTE performance

➢ Other Models

Apart from the logistic regression I have also implemented Random Forest and Decision tree algorithms to compare the model performances. Since the over sample data which was used for SMOTE is used for both this model.

Results of Decision Tree:

```
Accuracy: 0.998605496813825
Precision: 0.9978393380996715
Recall: 0.999384225619733
F1 Score: 0.9986111843608811

Confusion Matrix:
[[56340   123]
 [   35 56804]]

Classification Report:
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	56463
1	1.00	1.00	1.00	56839
accuracy			1.00	113302
macro avg	1.00	1.00	1.00	113302
weighted avg	1.00	1.00	1.00	113302

Fig 4.12 Summary of Decision tree performance.

When the decision tree was applied in our oversample data, we got Accuracy and F1 score of 0.99 and both the fraud and non-fraud class resulted in the perfect score of 1 which is very good.

Results of Random Forest Algorithm:

```
The evaluation measures for Random Forest Classifier :

 Accuracy: 0.9999117403046724
 Precision: 0.9998240954106493
 Recall: 1.0
 F1 Score: 0.9999120399690381

 Confusion Matrix:
[[56453    10]
 [    0 56839]]

 Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56463
           1       1.00      1.00      1.00     56839

    accuracy                           1.00    113302
   macro avg       1.00      1.00      1.00    113302
weighted avg       1.00      1.00      1.00    113302
```

Fig 4.12 Summary of Random Forest performance.

When the Random Forest algorithm was applied in our data, we got Accuracy and F1 score of close to 1 and both the fraud and non-fraud classes resulted in the perfect score of 1.

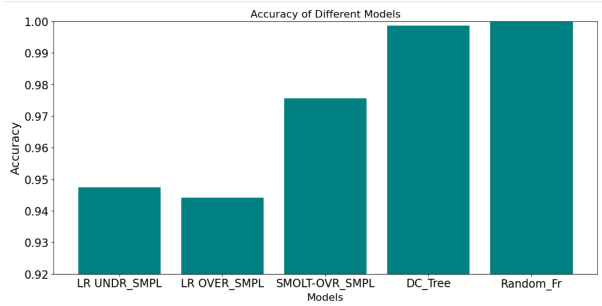


Fig 4.13 Accuracy Comparisons of different Models

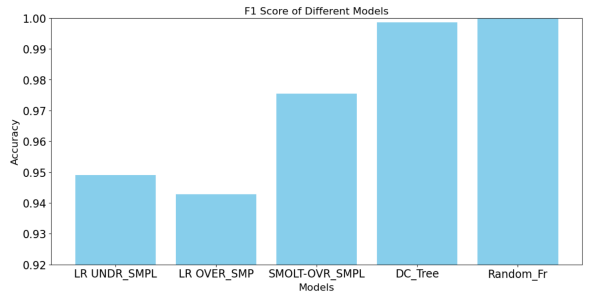


Fig 4.14 F1 Score Comparisons of different Models

Conclusion:

Time series Model.

In the research of Cocoa Prices analysis, we saw that ARIMA model performed the best in our case with the lowest RMSE Value of all i.e. 598.90 and forecast results were also close to the actual value which proves that ARIMA is best fitted model for our study. The other models like exponential smoothing by Holt's Method or Naïve model performance resulted in the high RMSE value which is considered as a poor performance. Although we can further improve the ARIMA model and bring down the RMSE score by doing double differencing in the data and making the series more stationary or trying different combination of p & q values.

In future this model can be definitely used to predict the future prices of the cocoa beans which will benefit the companies' investors as well as the shareholders to know how well the company is going to perform in future.

Logistic Model

The Logistic model gave the best results after applying SMOLT technique with the F1 score of 97.54 % and this is evident in the Fig 4.13 & 4.14 by analysing the F1 scores and accuracies of the different models performed. The visualisation shows that the F1 score instantly increased on performing the oversampling using SMOLT technique. We performed oversampling using the pythons sklearn library as well but the imblearn package proved to be more helpful in this case. Further we had also used Random Forest and Decision tree regression to see the performance and it turns out that both these models outperforms the logistic regression.

In future we can build a web application that will consume this model and we can evaluate it more by feeding the data in the frontend and building an alert mechanism that will notify us whenever the F1 score goes below 40% threshold.

REFRENCES:

[1] C. Chatfield, Time-series forecasting. Chapman and Hall/CRC, 2000

[2] Sirisha, Uppala & Belavagi, Manjula & Attigeri, Girija. (2022). Profit Prediction Using ARIMA, SARIMA and LSTM Models in Time Series Forecasting: A Comparison.

[3] Learning from Imbalanced Data Sets by Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, Francisco Herrera

[4] Mohammed, Roweida & Rawashdeh, Jumanah & Abdullah, Malak. (2020). Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results.

Links:

1)https://www.kaggle.com/code/prashant111/complete-guide-on-time-series-analysis-in-python

2) https://analyticsindiamag.com/quick-way-to-find-p-d-and-q-values-for-arima/

3)https://www.kaggle.com/code/hobeomlee/classification-using-smote-over-sampling

4) https://www.kaggle.com/code/chanchal24/credit-card-fraud-detection