**Results**

In this section we present results and analyse them**.**  
  
  
Below we used sales per customer data to forecast actual versus predicted sales.  
  
  
A graph with blue lines

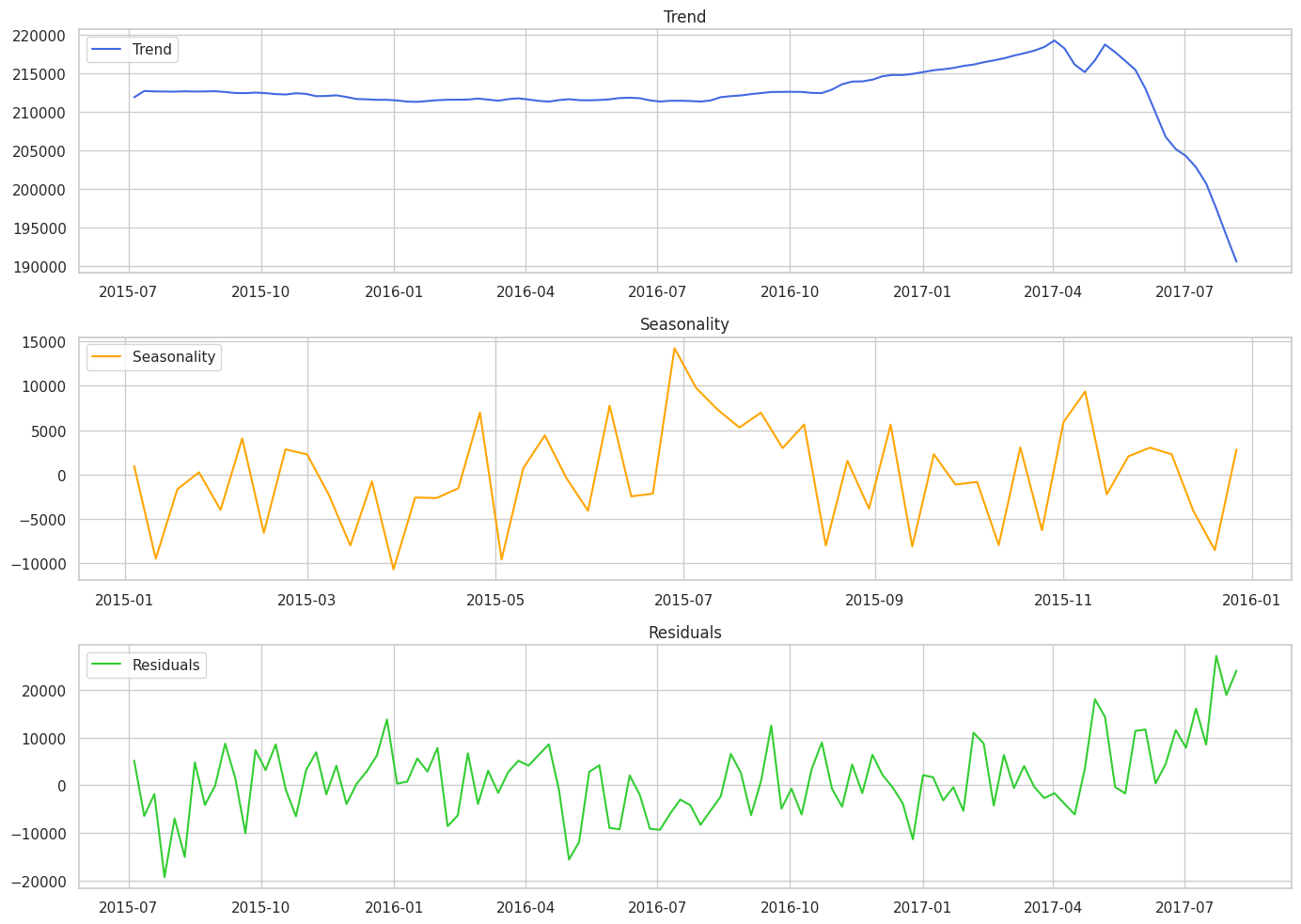
Description automatically generated

**SARIMA Forecast Vs. Actual Sales**

The 'Train' data, shown in the blue graph, falls under the sales per customer. This area is used by the SARIMA model to learn about the data's pattern. The values of 'Test', the symbol for the real sales data, which is used to validate the forecast, are shown in red, and none of these values are used during the training. One of it is 'SARIMA Forecast', which is represented by the green line; it is the prediction of model’s sale. The model seems to follow the trend of the test data but towards the end it did not capture the high variability. The SARIMA model in short-term forecasting, through the training phase, can follow the overall trend in the data, but in the test, it is not able to predict the volatility. The variance between the forecasted and actual sales during the test period suggests an uncertainty around the model and its precision, the model may require further optimization and the inadequate explanatory variables, or a different modeling approach may be required to draw a more accurate picture of the abnormal sales pattern shift. For SARIMA model parameter optimization, we are planning to make these parameters coherently compatible to the recorded sales spikes and dips, which in turn is expected to improve the forecast accuracy regarding both channeling seasonal dynamics and recording unusual events. Our model has been recognizing the option of bringing in factors outside as the ARIMAX method may happen and could involve the effort being made in marketing, trend, or events that has relationship with the sales. We researched into the most probable reasons for these fluctuations to know whether they are mere anomalies or trends that have just emerged, or to otherwise make appropriate changes to the forecasting approach. This approach will provide us with the ability to carry out multiple decision-making scenarios related to sales to avoid the interference of unpredictable trends. that behave in the market. In the next step we forecasted weekly sales using SARIMA as shown in the Fig.43

A graph with a line graph

Description automatically generated

Weekly sales Forecasting using SARIMA  
The train data, shown in blue, provides the historical context for the SARIMA model's learning process, indicating stable sales with slight fluctuations over time. The test data, in red, is essential for evaluating the model's predictive capability and is clearly distinguished from the training set to assess model performance on unseen data. The SARIMA forecast, plotted in green, extends beyond the historical data to predict future sales, with the shaded area representing the confidence interval of the predictions. The assumed prediction at the end turns out to be the average of testing data and tends to neglect the prominent variations, hence, the model requires upgrading to effectively accommodate the lumpiness in test data. The width of the forecast’s confidence interval increases with time, indicating that as the model predicts more in time, its predictions manage to match with less accuracy. We used seasonal decomposition as shown in the Fig.26 to better understand the trend, seasonality, and residuals. 

**Seasonal Decomposition**

The decomposition reveals insights in the structure of sales data, helping to understand how components like trend and seasonality, residuals contribute to the overall sales patterns. Trend (Royal Blue) component captures the underlying trend in the sales data, smoothing out the short fluctuations to highlight long term pattern. Seasonality (Orange): The periodic influence demonstrates a peculiarity or cyclicity of the data in accordance with the settings that have been made in advance, in this case, annually (52 weeks). This visualization helps spotlight the times in the year with steady, growing, or reduced sales. Residuals (Lime Green): Residual refers to the rest of the data that the model fails to explain. Optimally residuals should be randomly distributed around zero with no structured pattern, which suggests that model has successfully identified trend and seasonal characteristics. Original Data (Gray): The week's sales data are also plotted acting as a reference; hence the effectiveness of this decomposition's trend and seasonal components isolation can be examined. In the next step we forecast actual versus predicted weekly sales as shown in Fig.27.

A graph with colorful lines

Description automatically generated

**Actual versus Predicted sales**

These factors (blue line) serve as a baseline for the model (conveying actual past sales situation) with the purpose of informing of the forecast. The predicted training data indicates the model has learned the existing sales patterns precisely as the dashed red line is coincided with actual training data accurately. In a few places, predicted test data (dash green line) and the real data (solid orange line) differ, as for example, for sharp upward or downward peaks of the real sales statistics, thus showing where a model can be corrected in further work. In the initial phase the model forecasts coincide exactly with the actual sales, which implies that the historical data is quite stable. Inside the last phase, during the testing period, the model is not able to predict the extreme fluctuations, and this may be caused by unmodeled factors, which may be affecting sales or due to the limitations of the method used. We also forecasted the sales with Safety Margin and Reorder Threshold in the fig.28.

A graph showing a graph of sales

Description automatically generated with medium confidence

**Actual vs Predicted Sales with Safety Margin and Reorder Threshold**

The Actual Demand (blue line) stands for the sales history that is used in making special inventory planning decisions. The forecasting model does rather well in terms of 'Predicted Demand' (dashed red line), closely following the actual sales for much of the period, indicating good calibration.  
The Safety Stock (solid green line), however, remains steady giving a buffer against fluctuations in demand or any other supply problems. The 'reorder point' (dashed orange line) is under the busy ups and downs of actual demand, meaning that the refilling strategy is cautious and centered on avoidance of stock outs. The stock on some peaks is often higher than the safety stock and this leads to moments that the risk of stockout is increased, perhaps this might need adjustments in safety stock levels to follow the nature of demand as sudden increase in sales won't manage to result in shortage of stock.

**Sales Forecasting**The following table.5. shows results presented for various regression models used in sales forecasting exhibit diverse levels of performance.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Mean Absolute Error | Mean Square Error | RMSE | R^2 Score |
| Linear Regression | 0.001101 | 0.000003 | 0.001711 | 1.000000 |
| Gradient Boosting | 0.529301 | 1.287208 | 1.134552 | 0.999870 |
| Lasso Regression | 0.730684 | 0.984418 | 0.992179 | 0.999913 |
| Ridge Regression | 0.093759 | 0.024291 | 0.155856 | 0.945586 |
| ElasticNet | 15.125076 | 474.285200 | 21.778090 | 0.999925 |
| AdaBoost Regressor | 15.757847 | 423.601955 | 20.581593 | 0.999801 |

**A comparison of results presented for various regression models.**

• The R^2 score equal to 1.000000 obtained by Linear Regression model, indicates the model fit the best. The Mean Squared Error (MSE) and the Mean Absolute Error (MAE) are very close to zero indicating that nothing is left to perfect the model. Conversely, practical situations are not so perfect as to offer metrics of such magnitude, therefore, the high value of these metrics point to a case of overfitting or data leakage.

• Also, the Gradient Boosting Regressor potential has an almost perfect R^2 score of 0.999913 with a higher range of errors it is the MSE of 1.237910 and MAE of 0.529953 which are greater than with Random Forest. This means that the model is very reliable but with a little kerfuffle in terms of sales forecasts.

• For the Lasso Regression model with R^2 mean equals 0999870 which is nearly equal to 1 indicating that the model fits the data extremely well. The Mean Absolute Error of 0.730684 and Mean Squared Error of 0.984418 shows the model variation. This indicates the model is performing very well in predicting target variable.

• Ridge Regression model R^2 score equal to 0.945586 which is less than the Lasso Regression indicates that this model does not fit well as Lasso and Linear regression. Whereas the values MSE of 0.024291 and MAE of 0.093759 indicate that it provides more accurate prediction than the Lasso Regression.

• In ElasticNet the R^2 value equals 0.999925 which nearly equals 1 indicating the model perfect fit. However. MAE 15.125076 and MSE 474.285200, which are higher than the Linear, Lasso and Ridge regressions indicate that the ElasticNet has larger prediction errors.

• AdaBoost Regressor with R^2 value 0.999801 indicates the model perfect fit and MAE 15.757847 and MSE 423.601955 just like the ElasticNet has larger prediction errors.

The ensembles (RF, GBM, LR, RR, ElasticNet and AB) turn out to be reliable methods of sales prediction, making them conceivable for strategic decision-making based on forecasting. Their ability can account for non-linear relationships and interactions between variables leading to more reliable sales predictions. Simple models like Linear Regression has good fits, may not capture sales data complexity, while ensemble methods can be highly effective for capturing complex data of sales. Similarly, it is necessary to verify these models tests on overfitting and validation of it on an in discrete dataset to confirm their generalization.

To study outliers and to identify the areas in which the actual model performance should be improved we plotted a scatterplot in Fig.29. shown below.

A graph of a graph

Description automatically generated with medium confidence

**Actual vs Forecasted Sales**

The scatter plot depicts the comparison between the real and forecastable sales, with most data dots located close to the "perfect prediction" line, that is the dotted line. Such coordination implies the forecasts, though not spot on, are highly correlated with the actual sales. The uniform distribution of points along the range of values also demonstrates stable performance of the model in terms of accuracy, with deviations noticeable in the higher sales ranges and, probably, with these, the space for model improvement. Finally, studying these outliers could serve as the lens for identifying the areas in which the actual model performance should be improved.

From table.5 below we can see that the difference between the actual ordered items and the forecasted order items total is minute indicating that the predictive model performance in estimating the total order items.

|  |  |  |
| --- | --- | --- |
|  | Actual Order Item Total | Forecasted Order Item Total |
| 0 | 175.99 | 175.990391 |
| 1 | 245.00 | 244.999562 |
| 2 | 244.90 | 244.897846 |
| 3 | 251.98 | 251.980938 |
| 4 | 107.97 | 107.970609 |
|  |  |  |

**Comparison of Actual Order Item Total and Forecasted Order Item Total.**

**Late Delivery Prediction**The following table.7. Below shows the results of machine learning models used in late delivery prediction tasks to indicate varying degrees of performance across different algorithms.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 Score | ROC AUC |
| Logistic Regression | 0.975146 | 0.956884 | 0.9999798 | 0.977871 | 0.975164 |
| Decision Tree | 0.953726 | 0.959606 | 0.955993 | 0.957796 | 0.955021 |
| Random Forest | 0.972321 | 0.957944 | 0.993209 | 0.975258 | 0.978275 |
| K-Nearest Neighbors | 0.943257 | 0.928645 | 0.971323 | 0.949505 | 0.967172 |
| AdaBoost | 0.974555 | 0.956869 | 0.998689 | 0.977332 | 0.971456 |

**A comparison of metrics produced by all the models applied.**

• Logistic Regression which revealed a very high accuracy score (97.51%), and precision (95.6884%) nearly achieved a perfect ROC AUC score (97.5164%). Its F1 score (97.7871%) that balances the precision, and the recall is also high since this shows the model’s successful classification.  
  
• KNN algorithm yielded accuracy lower (94.3257%) against other approaches. The precision (92.8645%) and F1 score (94.9505%) were lower, and this might mean that the method might have slight misclassification problems. The 96.7172% ROC AUC score indicates a high capacity for differentiating between the classes, yet it is lower than the Logistic Regression model.  
  
• Decision Tree has accuracy of 95.3726% which is comparatively lower than the Logistic Regression. The precision of 95.9606% and F1 Score of 95.7796% indicates it might perform well in classification than the KNN.

• Random Forest has accuracy of 97.2321% which almost equals the Logistic Regression. The precision of 95.7944% and F1 Score of 97.5258% shows the model successful classification.

Adaboost with an accuracy of 97.4555%, precision of 95.6869%. ROC AUC (97.1456%) F1 score (97.7332%) indicating that the model performs well in classification and has low false positives.

The AUC scores for all models are above 0.9, which implies that the models being evaluated have good discriminatory ability. To conclude this, we checked the models against a separate validation set, performing cross-validation, adding techniques for mitigating overfitting like pruning (to decision trees) and regularization (for logistic regression). In conclusion, even though ensemble methods show striking results, it is vital to check this is not because of overfitting. All the regressions exhibit a good balance between all evaluating metrics being considered.

A graph of a curve

Description automatically generated 1. The plot shows the Receiver Operating Characteristic (ROC) curves for different techniques such as Logistic Regression, K-Nearest Neighbors, Decision Tree, Random Forest, Gradient Boosting, AdaBoost, XGBoost, and LightGBM regarding multi-class classification for the prediction of late deliveries.

2. The models have obtained an AUC score of 1.00 for all data sets, which sends the clear message that the models have been able to classify the data with incredible precision. This conclusion draws from the fact that the models can categorize all positive instances as correctly as they have no falsely positive classifications.

3. The performance of various models in this assessment is not distinguishable whereas the ROC curves for all models are overlapping. This may indicate that the way ROC curve has been computed or plotted is incorrect, as it is unusual that any real-world models will indeed achieve a perfect AUC of 1.00, unless the models are totally consistent in the studied scenario.  
  
4. Being that the ROC values are near perfect, it will be worth an implementation of exploration into the way the data was used and if the models are overfitting, if the training and test sets were leaking and so on, also examination of the situations that ROC AUC is calculated for a multi-class, but ROC AUC scores are typically used for binary classification to rule out any chances of miscalculations. . The following graph and table shows final results

A close-up of a graph

Description automatically generated

**Actual Delivery Status Distribution & Forecasted Delivery Status Distribution**

|  |  |  |  |
| --- | --- | --- | --- |
| Index | Customer Id | Actual Delivery Status | Forecasted Delivery Status |
| 0 | 6109 | Late delivery | 1 |
| 1 | 12041 | Late delivery | 1 |
| 2 | 4271 | Advance shipping | 0 |
| 3 | 3521 | Late delivery | 1 |
| 4 | 12073 | Advance shipping | 1 |

**Actual Delivery Status Distribution & Forecasted Delivery Status Distribution**

The variance between forecasted and actual statuses proves the output of the model's accuracy in predicting delivery outlook, especially for advanced shipment and overdue delivery. Unrealities between observed and predicted statuses prompts more analysis to advance the successfulness in delivery prediction and increase customer satisfaction. An analysis of 'Real Shipping Status' and 'Predicted Shipping Status' output shows that late deliveries are displayed the maximum time in the data set, and the order in this case is advance booking, delivery, delivery on time, and cancellation. The crashed image between the forecast and actual attendance leads to the model’s effectiveness in portraying distribution of availability. Nevertheless, the singular prediction accuracy in each must be assessed to consider the specificity of this prediction. Upgrading the forecasting model for increasing the prediction precision can result in an operation that is customer-oriented and efficient in terms of the delivery business.

**FraudDetection**. The following Table shows results for the classification models applied to fraud payment prediction illustrating varied performances across different metrics.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1 Score | AUC-ROC |
| Logistic Regression | 0.9929 | 0.9860 | 0.9999 | 0.9929 | 0.9985 |
| Random Forest | 0.9986 | 0.9978 | 0.9994 | 0.9986 | 1.0000 |
| Gradient Boosting | 0.9902 | 0.9807 | 1.0000 | 0.9902 | 0.9913 |
| K-Nearest Neighbors | 0.9923 | 0.9848 | 1.0000 | 0.9924 | 0.9984 |
| Decision Tree | 0.9986 | 0.9980 | 0.9991 | 0.9986 | 0.9986 |
| XGBoost | 0.9902 | 0.9806 | 1.0000 | 0.9902 | 0.9964 |

**Results for the classification models applied to fraud payment across different metrics.**

• Logistic Regression has a high-accuracy (99.29%) as well as a high precision (98.60%), which is outperformed only by recall (99.99%) and gives us an F1 score of 99.29%. The AUC-ROC score of 0.9985 depicts that the model is capable of distinguishing between fraudulent and legitimate transactions with a near-perfect level of accuracy. This model combines the capability for accurate predictions with the ability to recognize most of the fraudulent ones.

• Random Forest attains nearly perfect scores in all metrics, 99. 86% accuracy, F1 score and precision of 99. 78% and the recall value of 99. 94 respectively. The value of 100% for the AUC-ROC shows a good compromise between sensitivity and specificity in fraud detection that highlights the accuracy of the fraud detection model.

• Gradient Boosting gives rise to a slightly less accurate F-score of 99.02% and a 98.07% precision but a fully recallable score of 100%. The F1 score resembles accuracy, and the AUC-ROC score is 99.13%, showing excellent performance without missing a single fraud transaction whether is fake or legitimate.

• While KNN has the same accuracy and F1 measure score as the other two, its precision is 98.48% and its recall is a perfect 100%. The AUC-ROC score of 99.84% gives evidence of robustness of KNN in fraud detection, so that all cases of fraud will be captured by KNN.

• Decision Tree like Random Forest is 99.86% accurate and has an F1 score of 99.80% and it almost shows 100% precision, but it has 99.91% recall. The AUC-ROC score of 99.86% for this method can be considered sufficient for discriminability. Despite this, it is lower than that of Random Forest and Logistic Regression.

• XGBoost mimics Gradient Boosting both in accuracy (99.02%) and F1 score (99.02%), featuring precision measuring 98.06% and having recall equal to 100%. AUC-ROC score of 99.64% acts as a gauge of exceptional performance of the tool in segregating fraudulent from genuine transactions.To compare models we used Confusion matrix, precision recall curve ,cumulative Gains curve and calibration curve as shown in the Figures .34.35.36 Below.

A blue squares with white text

Description automatically generated

**Confusion matrix for comparison across different models applied.**

The confusion matrix shows the performance of classifiers that are binary i.e., K-Nearest Neighbors, Decision Tree, and XGBoost. K-Nearest Neighbors has a strong analytical capability, and it is the best in the negative class identification with truer and increasing negative and false positive and negative variations. Further, Decision Tree model provides like the Sensitivity vs Specificity balance among the false negatives more. In contrast XGBoost exhibits a high true positives rate with very high false positives that signifies a discarded trade-off between specificity of negative cases as positive. Eventually, all models show statistically significant values of coefficient of determination with P value of less than 0.05.

A graph with different colored lines

Description automatically generated

**Precision -Recall curve for Multiple models**

The Graph depicts the multiples models which are used in fraud analytics. One could remark that the Logistic Regression, Random Forest, K-Nearest Neigbhor, Decision Tress, and XGBoost score 1.00 AUC, which means perfect precision-recall balance. This means best performance in all conditions under both measures of precision and recall. The next to follow is Gradient Boosting with an AUC of 0.99, which is an exceptional accuracy that is slightly lower than precision and recall. When such a classifier yields a high AUC score, especially for the problems of fraud detection, this might be because of a carefully defined problem space or of the existence of a problem with data or model validation, like data leakage or overfitting.

**A graph of a number of graphs

Description automatically generated with medium confidence**

**Cumulative Gains Curve for Multiple models**

The cumulative gains curve is the performance of a group of predictive models combined. The models Logistic Regression, Random Forest, K-Nearest Neighbours, and Decision Tree are very competitive, with 1.00 Gini coefficients and rapidly climbing to their peaks at the graph’s top. This means their skill to catch the large of positive cases at an early stage. XGBoost and Gradient Boosting have Gini coefficients of 0.99 and 0.98 respectively and thus perform great also helping other leading models catch up. The vertical jumping points of all curves imply a strong degree of accuracy in the models in classifying positive cases when pursuing targeted marketing and risk rating activities. Nevertheless, the exceedingly high scores may engender even closer examination for the sake of detecting possible though non-existent overfitting or data leakage.

A graph of a graph with different colored lines

Description automatically generated

**Calibration Curves for Multiple Models**

The graph shows the calibration curves for different predictive models, like Logistic Regression, Random Forest, Gradient Boosting, K-Nearest Neighbors and Decision Tree, XGBoost. This chart shows the difference between observed and predicted probabilities for each model. In an ideal situation, the model needs to fit with the "Perfectly calibrated" line. However, most of the models deviate from it and have different points where they are not in line with the "Perfectly calibrated" which shows differences between the predicted probabilities and real outcomes. The Decision Tree method appears to be a perfect fit for calibration at the initial stage, but then it significantly drifts away, unraveling inaccurate predictions. Logistic Regression is the best model in terms of bias as it stays close to the diagonal line while XGBoost starts to deviate away from the line with higher predicted probability. Briefly, the presented models seem to perform better than that at a particular level of probability and others fail to do so in part of their forecasts. This is important information for risk assessment and decision-making procedures. The following Table.8. shows the output of applying different models to identify fraud across individual customers.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Customer Order ID | Actual Fraud | Predicted Fraud (Logistic Regression) | Predicted Fraud (Random Forest) | Predicted Fraud (Gradient Boosting) | Predicted Fraud (K-Nearest Neighbors) | Predicted Fraud (Decision Tree) | Predicted Fraud (XGBoost) |
| 48090 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 320949 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 120774 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 122605 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 83881 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 121848 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 144481 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 145764 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 210725 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 134653 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |

**shows results of Multiple Models applied to identify fraud in Customer orders.**

The outcomes of running different machine learning approaches to identify fraud in customer orders indicate a high precision accuracy all over the models. All models for instance, report and predicted fraud statuses for Customer Order IDs 48090 and 320949 show a unanimous agreement, suggesting that the models are very effective in prediction. Surprisingly, for Customer Order ID No: 134653 the predictions of both models differ, and this confirms the fact that fraud detection is not always straightforward since it often involves complex patterns. Model choice also becomes critical, depending on the data characteristics and business outcomes of false-positives and -negatives. The fact that models like Logistic Regression, Random Forest, and Gradient Boosting perform consistently across majority of the predictions might indicate their aptness for the said job. But the situations where the model misidentification like Customer Order ID 134653 happens from time to time demonstrate the critical significance of working on the model’s improvement.

**RFM analysis**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Order Customer ID | Recency | Frequency | Monetary | F\_score | R\_score | M\_score | Customer Segment |
| 1 | 793 | 1 | 472.45 | 1 | 1 | 2 | Need Attention |
| 2 | 137 | 4 | 1618.66 | 2 | 3 | 3 | Need Attention |
| 3 | 230 | 5 | 3189.20 | 2 | 2 | 4 | Other |
| 4 | 381 | 4 | 1480.71 | 2 | 1 | 3 | Need Attention |
| 5 | 458 | 3 | 1101.92 | 2 | 1 | 2 | Need Attention |
| … | … | … | … | … | … | … | … |
| 20753 | 1 | 1 | 161.87 | 1 | 4 | 1 | Loyal Customers |
| 20754 | 1 | 1 | 172.66 | 1 | 4 | 1 | Loyal Customers |
| 20755 | 1 | 1 | 314.64 | 1 | 4 | 2 | Loyal Customers |
| 20756 | 1 | 1 | 10.91 | 1 | 4 | 1 | Loyal Customers |
| 20757 | 1 | 1 | 34.98 | 1 | 4 | 1 | Loyal Customers |

The RFM analysis helps us to understand the customer lifetime value. From table.11 below we can see the customer segmentation based on RFM. The F\_Score, R\_Score and M\_Score are the percentile given based on the Recency, Frequency, Monetary. For the Order customer ID 1 their F\_Score and R\_Score are 1 indicating that they purchase less

Frequently and have less potential of repeating purchases. However, the M\_Score of 2 which means they make an average monetary value whenever they purchase. This customer segments as Need Attention customers which can be done by providing more offers and discounts. For the Order customer ID 3 The F\_Score and R\_Score indicate that they are average at purchasing the and frequency of their orders. Whereas the M-Score of 4 indicates that when they place an order, they pay a hefty amount. Hence, these customers are segmented as Other. For Order Customer\_ID 20753 they are segmented as Loyal Customers, because they purchase more often though their Monetary value is 1.

**RFM analysis and customer segmentation**