BT4012: Fraud Analytics

SEMESTER 1, ACADEMIC YEAR 2023/2024

Group Project: Fraud Supply Chain Analytics

Github: https://github.com/teresasee/BT4012-Supply-Chain-Fraud

Name	Matriculation Number
Calvin Septyanto	A0244491A
Chen Jia Wei	A0233484A
Liew Wei, Brian	A0239041M
Seah Ding Xuan	A0240134X
See Jia Wei, Teresa	A0240610Y

TABLE OF CONTENTS

1. Problem Background 4
    1.1 Project Objectives 4
    1.2 Dataset Description 4
2. Exploratory Data Analysis 5
3. Data Preprocessing 5
    3.1 Removing NA Columns 6
    3.2 Preventing Data Leakage 6
    3.3 Removing Other Redundant columns 6
    3.4 Train-test split & Group Shuffle Split 6
    3.5 Aggregation of rows 7
    3.6 Scaling 7
    3.7 Encoding Categorical Columns 7
    3.8 SMOTE and random undersampling 7
4. Feature Selection 8
5. Model Training & Evaluation 9
    5.1 Model Training 9
    5.2 Model Evaluation 9
    5.3 Hyperparameter Tuning 9
    5.4 Cross-Evaluation 10
6. Evaluation 10
    6.1 Model Results & Evaluation 10
    6.2 Advantages and Limitations of Chosen Models 11
    6.3 Integration of Fraud Detection Model into Business Processes 12
        6.3.1 API Development 12
        6.3.2 Transaction Monitoring System & Alert System 12
        6.3.3 Feedback Loop 12
7. Conclusion 12
    7.1 Areas of Improvements 12
        7.1.1 Quality of Data 12
        7.1.2 Domain Experts Knowledge 12
        7.1.3 Explainable ML 13
    7.2 Generalizability & Feasibility of Analysis 13
    7.3 Final Words 13
8. Appendix 14
    8.1 Charts for Exploratory Data Analysis 14
        8.1.1 Fraudulent Rate by Payment Type 14
        8.1.2 Fraudulent Rate by Order Time (Month, Day, Hour) 14
        8.1.3 Fraudulent Rate by Late Delivery Risk 14
        8.1.4 Fraudulent Rate by Customer Segment 15
        8.1.5 Fraudulent Rate by Shipping Mode 15

8.1.6 Fraudulent Rate by Market 15
8.1.7 Fraudulent Rate by Order Country 16
8.1.8 Fraudulent Rate by Department Name 16
8.2 Feature Importances 16
8.3 Cross Evaluation of Train-Test Loss 17
8.3.1 Logistic Regression 17
8.3.2 AdaBoost 18
8.3.3 Gradient Boosting Classifier 19
8.3.4 LightGBM Classifier 20
9. References 21

1. Problem Background

In the contemporary business landscape, e-commerce sales have ramped up considerably over the years, accounting for over USD $5 trillion in 2022 (Chevalier, 2022), thereby placing a huge demand on the e-commerce supply chain systems. To provide satisfactory services and thrive in such fast-paced industries, supply chain firms have to take many precautionary measures to mitigate external risks. One of these areas of risks faced by supply chain companies is fraud, including but not limited to: third-party retailer fraud, customer fraud, and fraudulent billing (Dutta et al., 2019). According to Coppola (2023), e-commerce losses to online payment fraud were estimated at 41 billion U.S. dollars globally in 2022, up from the previous year, and this figure is expected to grow further to 48 billion U.S. dollars by 2023. Due to the sheer number of daily transactions, scale of the global systems, and the complexity of the operating environment involving numerous stakeholders, it can be difficult to pick up on these fraudulent activities (KPMG, 2020). Fraud risks can result in significant losses, such as decreased reputability, increased processing times for goods, and substantial financial losses from insurance, product recalls, litigation costs, and reduction in market share (KPMG, 2017).

1.1 Project Objectives

In our project, we will be attempting to develop a machine learning pipeline to aid in reducing e-commerce supply chain fraud, particularly customer fraud and potential fraudulent billing activities. We wish to achieve the following benefits via this project:

1. Extract out and pre-process transaction details useful in determining fraudulent transactions
2. Train a supervised machine learning model capable of detecting fraudulent transactions
3. Flag out any suspicious orders when the order is placed with the trained models
4. Identify features that may contribute to the orders being flagged as fraudulent

1.2 Dataset Description

The dataset we will be using is provided by DataCo Global, which can also be found on Mendeley (Constante, 2019). The dataset, comprises 180,520 rows and 53 columns, contains transaction information of a supply chain operator, summarised as follows:

Feature Type	**Description**
Transaction Details	Details the transaction date and payment method (e.g. cash, debit, transfer).
Shipping Information	Shipping details such as the number of days for shipping, shipping date, as well as the delivery status (e.g. late delivery, advanced shipping),
Product Details	Information about the product and its department, such as product category, department category, price, and product description.
Customer Details	Information about the person who made the purchase. This includes their name, their address (e.g. state, city, country, street)
Store Details	Data about where the purchase was registered. This includes the store's department, its location, and which market it caters to (e.g. LATAM).
Order Item Details	Information about the order date, order location and discounts applied for the order item. Includes order status, which indicates the current state of the order (e.g. completed, pending, potential fraud).

The dataset consists of a mix of date fields (shipping & order dates), integers (number of items purchased, number of days to ship), floats (discount applied, product prices) and strings (payment method, market, location, order status, product description, product link, name, address).

2. Exploratory Data Analysis

We will be conducting EDA on our *train set* to determine the important features for our predictors.

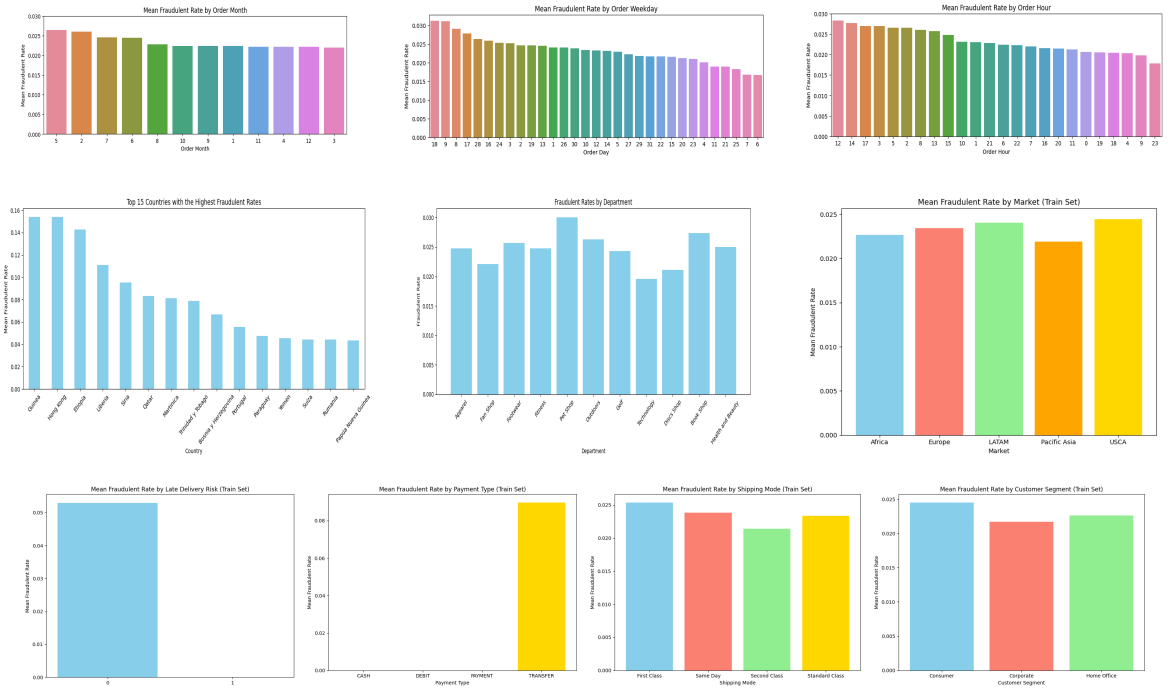

Figure 1: Visual Summary of Supply Chain Train Dataset EDA

We used the EDA to identify attributes in our dataset that are closely tied to the rate of fraudulent transactions. Starting with time-based attributes, 'order_month', 'order_weekday', and 'order_hour' provide insights into specific times when fraud is more prevalent. Following this, 'order_country' helps us spot which countries have a higher incidence of fraud. We then look at 'department_name', covering ranges from 'Department_Name_Apparel' to 'Department_Name_Health_and_Beauty', to understand fraud trends within various product categories.

Moving on to broader categories, 'market categories' such as 'Africa', 'Europe', 'LATAM', 'Pacific_Asia', and 'USCA' are selected to identify regional fraud trends. The feature 'late_delivery_risk' is noted for its reverse link to fraud occurrences. For payment methods, we can see TRANSFER' payment type has a high association to fraudulent transactions. However, it's ***important to clarify*** that not every transaction labelled as 'TRANSFER' or marked with a non-late delivery risk is fraudulent, avoiding any potential data leakage in our model. Shipping options including 'First_Class', 'Same_Day', and 'Second_Class' are chosen for their varied fraud rates. Lastly, we consider customer segments such as 'Consumer', 'Corporate', and 'Home_Office', as they show different levels of fraud risk. These features are chosen for their strong and clear connections to fraudulent activity, ensuring our model's effectiveness in fraud detection.

3. Data Preprocessing

Before we carried on with predictive analytics, our team performed exploratory data analysis to prepare our data for training.

3.1 Removing NA Columns

	index	0
46	Product Description	1.000000
43	Order Zipcode	0.862397
14	Customer Lname	0.000044
19	Customer Zipcode	0.000017

Figure 2: Percentage of NA In Dataset Columns

Product Description and *Order Zipcode* were removed due to their high percentage of NA counts.

3.2 Preventing Data Leakage

We found that all records that were fraudulent had *Delivery Status* of “Shipping cancelled”. We decided to remove the *Delivery Status* column to prevent data leakage, as further understanding of the dataset showed that shipping of these orders may have been cancelled as they were flagged as fraudulent. Additionally, it came to our attention that when *Delivery Status* is "Shipping cancelled", the corresponding *Order Status* will be "CANCELLED" or “SUSPECTED_FRAUD”. Since we will be removing “CANCELLED” orders that did not go through, all remaining “CANCELLED” orders will be fraudulent transactions, hence the feature needs to be removed to prevent any form of data leakage.

3.3 Removing Other Redundant columns

In our analysis, it is essential to focus on features that exhibit meaningful variation and contribute to our understanding of the underlying patterns or relationships. Thus, we have methodically removed certain columns. The 'Product Status' column was excluded due to its uniform value of 0 across the dataset, providing no variability or useful insights. The 'Product Image' column, containing only URLs to product images, was also deemed redundant since the product names, which we have retained, offer the necessary information.

Further, we dropped several ID columns including 'Department Id', 'Category Id', 'Customer Id', 'Order Customer Id', 'Product Card Id', 'Product Category Id', 'Cardprod Id', 'Order Item Id', and 'Order Id'. These columns are unnecessary because the descriptive names related to each ID are already part of our dataset, for instance, 'Department Name' instead of 'Department Id', which adds more context and value to our analysis.

Columns holding customer details such as 'Customer Email', 'Customer Password', 'Customer City', 'Customer Country', 'Customer Fname', 'Customer Lname', 'Customer State', 'Customer Street', and 'Customer Zipcode' were removed as well. These attributes either contained censored information or did not contribute meaningful data for our analysis. The columns 'Latitude' and 'Longitude' were also eliminated; while they offer precise location data, our analysis requires only general location information, such as 'order_country'.

Additionally, we eliminated specific order-related columns like 'Order State', 'Order Zipcode', and 'Order City'. Just like the 'Latitude' and 'Longitude' data, these details are too granular for our broader geographical focus. Lastly, financial columns such as 'Sales', 'Order Item Discount', 'Order Item Discount Rate', 'Order Item Product Price', 'Order Item Profit Ratio', and 'Product Price' were removed. These are all components of the 'Order Item Total' calculation and their exclusion is aimed at reducing multicollinearity and simplifying our dataset (Maaten & Postma, 2009).

3.4 Train-test split & Group Shuffle Split

We conducted a train-test split of 70-30, where 70% of the data was dedicated to training, allowing our models to learn from a diverse set of transactions. The remaining 30% served as an independent test set, to check the effectiveness of our model in identifying fraudulent activities not encountered during training. Since there may be duplicate order IDs before aggregating the rows, we will be using shuffle split

to split the unique IDs, preventing data leakage. In addition, we will be randomly shuffling the dataset during the process to prevent bias and improve model performance (Mishchenko et al., 2020).

***NOTE: This step is done before EDA**

3.5 Aggregation of rows

In the initial stages of our data analysis, we identified a noteworthy issue within our dataset—specifically, the presence of duplicate order IDs across rows. Multiple rows are associated with the same order ID, reflecting instances where customers had purchased multiple items within a single order. Recognizing the potential ramifications of this duplication, we became concerned about the risk of data leakage into the test set. The prospect of having rows from the same order present in both the training and test sets could compromise the integrity of our model evaluation (Nayak & Ojha, 2020). To mitigate this risk, we decided to aggregate the rows associated with each order involving merging rows that shared the same order ID, consolidating them into a single, representative row.

***NOTE: This step is done before EDA to prevent data leakage**

3.6 Scaling

In our data set, most of the continuous features are not on the same scale. For example, both the order_total and product_price range from [0: Inf], but they have different distributions and different minimum/maximum values. During the calculation of Euclidean distance to determine feature importance, features with larger values will naturally take a higher weight compared to that of smaller values, resulting in the neglection of features with smaller values. To prevent this from happening, we use ***min-max scaling*** to normalise all the numerical attributes, so that all features have an equal weight.

3.7 Encoding Categorical Columns

For the categorical columns, we used one hot encoding for the columns with lower cardinality and frequency encoding for the remaining.

Columns with one hot encoding: customer_segment', 'shipping_mode', 'payment_type', 'market'

For categorical columns with no ordinal relationship such as customer_segment', 'shipping_mode', 'payment_type' and 'market', one-hot encoding is employed to transform these variables into numeric input values suitable to be used for machine learning models. One-hot encoding involves the creation of binary dummy variables for each unique category in the column, and the presence and absence of each category is represented with ‘1’ and ‘0’ respectively.

Columns with frequency encoding: order_country’, ‘order_month’, ‘order_weekday’, ‘order_hour’

For the ‘order_country’, ‘order_month’, ‘order_weekday’ and ‘order_hour’ columns, we decided to conduct frequency encoding due to high cardinality to retain the information provided by the column while not compromising the high dimensionality of the dataset (Roy, 2023).

3.8 SMOTE and random undersampling

The dataset used also has a significant class imbalance, with far fewer fraudulent rows than non-fraudulent ones. This imbalance makes it difficult for models to effectively detect fraud. To address this, we use a two-pronged approach. First, Synthetic Minority Over-sampling Technique (SMOTE) was used to oversample the minority class, generating synthetic instances of fraudulent transactions. This approach increased the representation of the minority class in the training dataset, mitigating the impact of class imbalance (Sisodia et al., 2017).

Following SMOTE oversampling, we then used random undersampling to further improve the dataset. The undersampling process involves randomly removing instances from the majority class (non-fraudulent transactions) to ensure a proportional representation of both classes. In this case, it gives us a balanced dataset consisting of 10,000 rows each of both fraudulent and non-fraudulent transactions. By balancing the dataset in this manner, the model is provided with a more equitable training environment, enabling it to learn from a diverse set of instances from both classes (Kotsiantis et al., 2005).

4. Feature Selection

Before model training, we trained a basic logistic regression against our base dataset (after the initial drop of redundant and NA columns). The results of the logistic regression are then used as a benchmark to select our final set of features used for training our final fraud detection model. We used 2 methods of feature selection on the base dataset to get 2 different datasets: **Dataset A** - Lasso regression and **Dataset B** - manual feature selection.

Dataset A - Using Lasso regression, fitting the model with L1 penalty ensures unimportant features are penalised to 0 (Muthukrishnan & Rohini, 2016).

Feature selection is done by selecting only the coefficients which are non-zero. Notably, several features such as 'Consumer', 'Department_Name_Unknown', 'Department_Name_Pet_Shop', 'Department_Name_Fitness', 'Europe' and 'LATAM' have coefficients reduced to 0, suggesting a minimal contribution to the model. Removing these features ensures that the models focus on only more influential features, enhancing model interpretability and predictive performance (Zhang et al., 2021). The selected **95** features are shown in *the code*.

Dataset B - Manual selection was conducted based on our intuition of which features are important, or which features are highly correlated (*More details in EDA and Data Preprocessing section*).

For example, since shipping_date is highly correlated with order_date, we removed all the features extracted from shipping_date, including shipping_month, shipping_weekday, shipping_hour. The other columns removed are num_distinct_items, order_item_quantity, order_total, order_region, shipping_month, shipping_weekday, shipping_hour. Manual selection of features allows for the integration of domain knowledge, addressing correlations, prioritising relevant information to reduce dimensionality, leading to improved model interpretability and performance better suited for identifying fraudulent activities (Kazi et al., 2022). The **29** remaining features are:

'late_delivery_risk', 'order_country', 'Department_Name_Fitness', 'Department_Name_Golf', 'Department_Name_Footwear', 'Department_Name_Apparel', 'Department_Name_Outdoors', 'Department_Name_Fan_Shop', 'Department_Name_Technology', 'Department_Name_Book_Shop', 'Department_Name_Discs_Shop', 'Department_Name_Pet_Shop', 'Department_Name_Unknown', 'order_month', 'order_weekday', 'order_hour', 'Consumer', 'Corporate', 'Home_Office', 'First_Class', 'Same_Day', 'Second_Class', 'TRANSFER', 'Africa', 'Europe', 'LATAM', 'Pacific_Asia', 'USCA', 'Department_Name_Health_and_Beauty'

We used these datasets for training several basic machine learning models to select the dataset for future model training. The outcomes revealed that dataset B showed better performance in the regression task, as seen in the results table below.

	Model Name	Accuracy	Recall	Precision	Specificity	F1 Score	AUC
Dataset B	LogisticRegression	0.893508	1.000000	0.181131	1.000000	0.306707	0.959124
Dataset A	LogisticRegression	0.893715	1.000000	0.181419	1.000000	0.307121	0.958262
Base Dataset	LogisticRegression	0.893870	1.000000	0.181637	1.000000	0.307432	0.958001
Dataset B	RandomForestClassifier	0.917012	0.778022	0.190733	0.778022	0.306361	0.952555
Dataset B	LGBMClassifier	0.928401	0.668132	0.197917	0.668132	0.305374	0.956071
Dataset A	LGBMClassifier	0.936995	0.665934	0.221491	0.665934	0.332419	0.960470
Dataset A	RandomForestClassifier	0.931508	0.646154	0.201923	0.646154	0.307692	0.952864
Base Dataset	LGBMClassifier	0.936995	0.610989	0.210926	0.610989	0.313593	0.958535
Base Dataset	RandomForestClassifier	0.945900	0.439560	0.202020	0.439560	0.276817	0.948926

Figure 3: Performance of Datasets (Base Dataset/Dataset A/Dataset B) with Respective Machine Learning Models

Consequently, we made the strategic decision to employ dataset B exclusively for subsequent model training to reduce computational complexity without sacrificing model performance.

5. Model Training & Evaluation

5.1 Model Training

A variety of machine learning models are used for testing and evaluation to determine which models perform the best in fraud detection for our supply chain dataset. Models used include Extreme Gradient Boost (XGBoost), Light Gradient Boosting Machine (LGBM) (Upadhyay et al., 2021), Random Forest, Logistic Regression, Gradient Boosting Classifier, Multi-Layer Perceptron (MLP), Adaptive Boosting (AdaBoost), Support Vector Classifier (SVC), k- Nearest Neighbours and Gaussian Naive Bayes.

Using the train sets, each model will be trained on **Dataset B (Manual Selection)**. After which, we will be evaluating the respective models using the test set.

5.2 Model Evaluation

Metrics that were used to evaluate our supply chain fraud detection model include **accuracy, specificity, F1 score, recall, precision, and AUC** to ensure a well-rounded assessment of our machine learning model's effectiveness in tackling the intricacies of fraud detection. The results are compiled in a table to evaluate how well each model performs in detecting fraud in supply chain transactions, as shown in Figure 4 below:

	Model Name	Accuracy	Recall	Precision	Specificity	F1 Score	AUC
0	LogisticRegression	0.893508	1.000000	0.181131	1.000000	0.306707	0.959124
9	AdaBoostClassifier	0.898374	0.973626	0.185046	0.973626	0.310986	0.958052
2	GradientBoostingClassifier	0.897184	0.964835	0.182233	0.964835	0.306564	0.957206
8	LGBMClassifier	0.928401	0.668132	0.197917	0.668132	0.305374	0.956071
6	MLPClassifier	0.905156	0.870330	0.182573	0.870330	0.301829	0.955786
7	XGBClassifier	0.939118	0.586813	0.212749	0.586813	0.312281	0.955442
1	RandomForestClassifier	0.917012	0.778022	0.190733	0.778022	0.306361	0.952555
3	SVC	0.894026	0.989011	0.180578	0.989011	0.305395	0.948165
5	GaussianNB	0.894906	0.980220	0.180786	0.980220	0.305270	0.945785
4	KNeighborsClassifier	0.903241	0.912088	0.184938	0.912088	0.307521	0.943596

Figure 4: Performance of Dataset B with Respective Machine Learning Models

We chose **Recall and AUC** as the 2 main metrics that we will be using to evaluate our machine learning models for the reasons mentioned below:

Recall

Recall is a measurement of the model's ability to capture all positive instances in the dataset. In the context of supply chain fraud detection, having a high recall is important because it helps minimise false negatives - situations where the model fails to identify actual fraud instances. Missing fraudulent transactions can lead to financial losses as well as negative supply chain impacts (Jeni et al., 2013).

AUC (Area Under the ROC Curve)

In the context of supply chain fraud detection with imbalance datasets, AUC is particularly important as it is an aggregate measure on the ability of the model to differentiate between fraudulent and non-fraudulent purchases across different decision thresholds. A better discrimination between negative and positive instances can be reflected by a higher AUC. AUC is also valuable when class distribution is imbalanced such that of our dataset, where non-fraudulent transactions largely outweigh fraudulent ones as it is less sensitive to class imbalances compared to other metrics (Hancock et al., 2023).

Together, AUC and recall offer a balanced evaluation of the model's performance in the specific context of fraud detection.

5.3 Hyperparameter Tuning

From the results, we decided to select the top 4 performing models with the highest recall and AUC to conduct hyperparameter tuning. The top 4 models we selected are the AdaBoost, LGBM, Gradient Boosting and Logistic Regression classifiers.

For each model, grid search was performed on a set of possible hyperparameter values. For instance, hyperparameters combinations such as 'n_estimators', 'max_depth', 'min_samples_split', 'min_samples_leaf' and 'max_features' were used for the Random Forest model.

Using k-fold cross validation (Jung & Hu, 2015), grid search evaluates each combination of hyperparameters using the ROC AUC score (scoring='roc_auc') as the evaluation metric. In the case of our project, we used 5-fold cross-validation (cv=5), in which the train dataset will be split into 5 folds and the model will be trained and evaluated 5 times, using a different fold as the validation set each time. The grid search will identify the best set of hyperparameters, which will be used to train the final model for each algorithm.

	Model Name	Accuracy	Recall	Precision	Specificity	F1 Score	AUC
0	LogisticRegression	0.893508	1.000000	0.181131	1.000000	0.306707	0.959099
1	AdaBoostClassifier	0.906295	0.872527	0.184737	0.872527	0.304916	0.957163
3	LGBMClassifier	0.939118	0.591209	0.213662	0.591209	0.313886	0.956129
2	GradientBoostingClassifier	0.930265	0.628571	0.195355	0.628571	0.298072	0.953998

Figure 5: Final Results of Top 4 Models After Hyperparameter Tuning

5.4 Cross-Evaluation

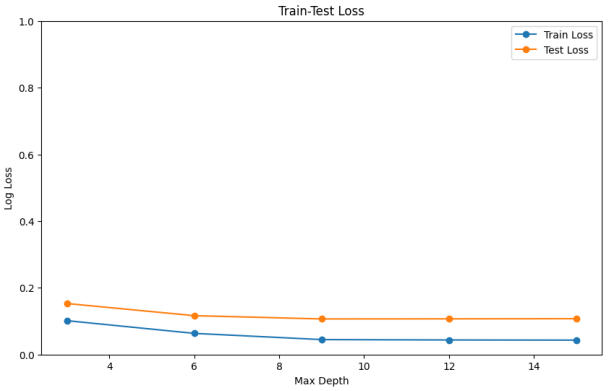

Figure 6: Graph showing train-test loss over max depth of LightGBM Classifier

We also analysed the logarithmic loss associated with some hyperparameters during the model's execution on both the training and test sets by graphing it. This examination aimed to identify any significant divergence in loss between the training and test datasets, allowing us to assess the potential for overfitting. In the above Figure 6, we can see that as the max depth of the classifier increases, the loss of both training and testing sets decreases. There are no signs of divergence where test loss increases while training loss decreases which implies there is no overfitting.

6. Evaluation

6.1 Model Results & Evaluation

After the hyperparameter tuning step and sanity checks that there is no severe overfitting for the models through cross-evaluation of the logarithmic losses between both train and test sets, we determined that the best model after all is still the logistic regression model (Figure 5). The following hyperparameters are determined to yield the highest AUC scores, namely: a regularisation strength 'C' of 0.1, with a bias 'fit_intercept' = True, a 'l2' penalty term to penalise model complexity and using a Newton-Conjugate Gradient Optimization 'newton-cg' algorithm (Wang et al., 2010).

The model boasts an AUC score of 0.959 and a recall of 1. This implies that the model is able to detect all the fraudulent cases, but at the expense of misclassifying some non-fraudulent cases, giving us a precision of 0.181. However, in the scenario of the supply chain fraud detection, our group determines that the

results are acceptable as we should attempt to pick up potential fraudulent cases, even if they are not. The confusion matrix (Figure 7) illustrates the above results.

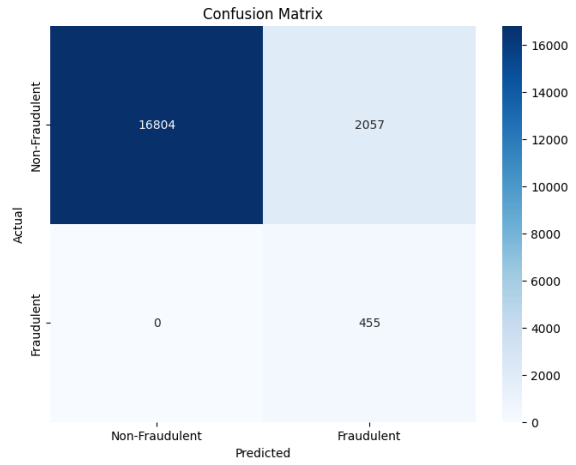

Figure 7: Confusion Matrix of the Logistic Regression Model

```
TRANSFER: 5.990233500128663
late_delivery_risk: -5.912987873424224
First_Class: 1.646195804519806
Second_Class: 0.6697868073189813
Department_Name_Technology: -0.5559713602910812
Department_Name_Outdoors: -0.47574689035116
Department_Name_Fan_Shop: -0.3790191240466647
Department_Name_Health_and_Beauty: -0.3674484938925861
order_month: -0.27924814038999674
order_country: 0.2693725684386112
```

Figure 8: Top 10 Coefficients of Logistic Regression Model

Our group does want to note that achieving 100% recall is not commonly seen. Therefore, we took a closer look at the coefficients of the logistic regression, finding the highest weights that may help us understand which features can be more important (Zhang et al., 2018). As shown in Figure 8, we can see that two of the features have extremely high coefficients, namely the TRANSFER payment method and the late delivery risk, which was also noted in our EDA step, where we see that all fraudulent classes to use TRANSFER payment and has a late delivery risk (potentially due to geographical factors). Given this specific dataset, it can explain why the logistic regression model performs the best as it will simply separate the transactions that use transfer payments and has a risk of late delivery. The other models exhibit similar behaviour, which falls under the appendix. Our group acknowledges that there may be cases of fraudulent transactions using other payment types or has no late delivery risk, which we will be further discussing in the future area of improvements.

6.2 Advantages and Limitations of Chosen Models

Top 4 Models	Description	Advantages	Limitations
AdaBoost Classifier	Adaptive boosting tree algorithm: Higher weights iteratively reassigned to incorrectly classified instances (Abraham et al., 2017).	- Adjusts the weights of misclassified data automatically. - Resistant to overfitting for datasets with low-noise.	- Sensitive to outliers and noisy data. - Performance heavily dependent on data quality. - Can be slower due to the sequential model building nature of the model.
LightGBM Classifier	Gradient boosting tree structure that expands vertically (Chen et al., n.d.).	- Handles large datasets well. - Performs well on unbalanced datasets e.g. fraud datasets. - Faster than other boosting algorithms	- Likely to overfit when dataset is small - Requires careful tuning of hyperparameters. - Complex algorithm, harder to interpret.
Gradient	Iteratively chooses	- High predictive power.	- Can be computationally expensive

Boosting Classifier	weights in the direction of the negative gradient of a loss function to minimise loss (Alexey & Alois, 2013).	- Supports different loss functions. - Robust to outliers and can handle heterogeneous features.	- Longer training times - Prone to overfitting without proper tuning.
Logistic Regression	Applies a logistic function to a linear combination of input features.	- Simple and easy to implement. - Good for probability estimation. - Outputs have a nice probabilistic interpretation. - Interpretable coefficients	- Limited to linear decision boundaries - Can struggle with complex relationships in data. - Not ideal for capturing interactions between features.

6.3 Integration of Fraud Detection Model into Business Processes

6.3.1 API Development

A pivotal step after development of the model involves the development of an Application Programming Interface (API) for our model, serving as the conduit for receiving transaction data and delivering predictions with agility, whether in real-time or as part of periodic batch processes.

6.3.2 Transaction Monitoring System & Alert System

The developed API should then be integrated into existing transaction monitoring systems to identify and address potential fraudulent activities quickly and efficiently within the operational framework of previous systems. The company should then align their alert systems with our fraud detection model. Transactions that are identified as potentially fraudulent trigger immediate alerts, which are then sent to a specialised team for in-depth investigation.

6.3.3 Feedback Loop

A continuous improvement strategy should also be adopted. The results of investigative processes, whether they confirm fraudulent activity or validate legitimate transactions, are systematically fed back into the model. This iterative feedback loop is essential for the continual refinement of our model, ensuring its accuracy and adaptability in an ever-changing landscape of fraudulent activity.

7. Conclusion

7.1 Areas of Improvements

7.1.1 Quality of Data

Looking over our model evaluation results (Figure 4), most of our models have a high recall close to 1, indicating their bias towards predicting the majority of the transactions as fraudulent, potentially erring at the side of caution at the expense of misclassifying legitimate transactions as fraudulent. Enhanced data quality can mitigate this by broadening the training dataset and enabling the model to learn from a more complex array of fraud characteristics, increasing our model performance and robustness: 1) Having a larger set of fraudulent transactions of varying types (e.g., those that use other payment methods). 2) Increasing the scope of the data, looking at supply chain fraud beyond the US to other countries and studying the geographical differences. 3) Collecting relevant features that can help detect fraud (previous customer purchases, review scores, etc.) 4) Adding dataset that details the transaction metadata (IP address, biometrics, and security methods used) to allow the model to uncover odd transaction patterns.

7.1.2 Domain Experts Knowledge

To improve our model's prediction capabilities in detecting supply chain fraud, it's important to gain knowledge from experts in supply chain management and logistics, who can provide insights on normal

transaction patterns, from typical order sizes to inconsistencies in the order-to-delivery timeline and common issues in shipping logistics, such as unusual routing of shipments. This set of patterns can help us develop better rules for identifying odd transactions.

7.1.3 Explainable ML

Another important issue to address is improving model interpretability using Explainable Machine Learning (XAI) techniques (Zhou et al., 2023). While our current model selection is aligned with transparency goals, further exploration into models with explicit interpretability, such as decision trees, can increase our understanding of the decision-making process. Additionally, there is room for further feature engineering that directly corresponds to the intricacies of supply chain fraud, potentially boosting the model's accuracy and interpretability. Using advanced techniques like SHAP (SHapley Additive exPlanations) for feature importance analysis and LIME (Local Interpretable Model-agnostic Explanations) for local interpretability can provide deeper insights into the model's behaviour at both global and individual transaction levels. Visualising decision boundaries offers an opportunity for clearer understanding of how the model differentiates different classes.

7.2 Generalizability & Feasibility of Analysis

From the start to the end, our project aims to develop a prototype model, with features that can be collected in real-time. These features that we have used in our preprocessing and modelling steps are features that can be obtained in all supply chain transactions (e.g., order country, time of purchase), likely to be recorded in the databases of the company. Subsequently, our models use these features to predict any potential fraud when the transaction is made, preventing fraudulent transactions from even going through in the first place. The model can be deployed at this stage in most supply chain companies similar to DataCo Global, and be continually trained and updated with new data coming in. Therefore, in this project, we have achieved a reasonably feasible and generalizable model used to predict fraud in supply chain operations. As mentioned in 7.1, there can be further improvements for better generalizability.

7.3 Final Words

This project serves to produce a fraud detection model contextualised for the supply chain transactions provided by DataCo Global. We utilised data modelling techniques to refine the data at hand and present a final model that can be used to detect fraud with relatively decent metrics. Integration of our tuned model can help us efficiently detect potential fraudulent cases, subsequently handed over to professionals for further investigation, reducing manpower concerns and achieving our project objectives.

Due to the nature of our dataset, we experimented with supervised models to detect fraud, and ultimately discovered the tuned logistic regression model to give us the best performance. Our group came to a conclusion that the logistic regression model we produced through different techniques has achieved a commendable result, measured appropriately with the recall and AUC metrics in an attempt to pick up many fraudulent transactions while keeping precision reasonable.

In hindsight, we also recognise the vast differences in supply chain operations, which our data may not fully encapsulate, making it difficult to apply to all shapes and sizes of supply chain fraud. There may be gaps in our data quality and quantity that we cannot resolve, and there are definitely areas of improvement that can be worked on in future work to enhance the deployment of such fraud detection models in specific use cases for supply chains all around the world.

We acknowledge that our project and other published reports on this dataset employ different data modelling methods. Similarly, Kim (2023) found that logistic regression was the most effective method for predicting fraud in the dataset. Data processing techniques however differed from ours in that they included the name and other columns without aggregating the rows. Recall scores across his models also showed exceptionally high results, which further reinforces our concern regarding the dataset.

8. Appendix

8.1 Charts for Exploratory Data Analysis

8.1.1 Fraudulent Rate by Payment Type

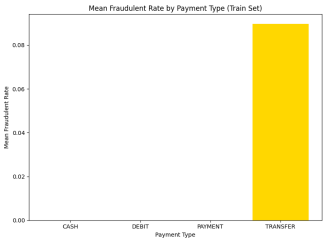

Figure 9: Bar Plot of Fraudulent Rate by Payment Type

As seen in the bar charts above, the "TRANSFER" payment type has a significantly higher mean fraudulent rate, close to 0.08, while the other payment types like "CASH", "DEBIT", and "PAYMENT" show almost negligible rates. Looking at the significant difference of fraudulent rate between “TRANSFER” and other payment types, we can infer that the payment type, especially "TRANSFER", is an important feature to consider in predicting fraudulent activities for our dataset. As explained in the *EDA section*, not all payment types under “TRANSFER” are fraudulent, so there is no data leakage involved.

8.1.2 Fraudulent Rate by Order Time (Month, Day, Hour)

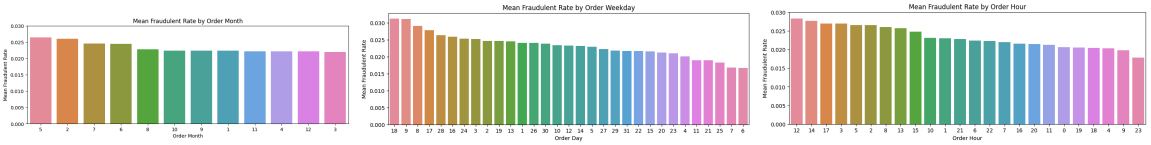

Figure 10: Bar Plot of Fraudulent Rate by Order Time (Month, Day, Hour)

When we look at the relationship between order time and fraudulent transactions, we need to find certain patterns related to the time the transactions occur and how often they are considered fraudulent from month, day, and hour granular breakdown. From our first bar plot on top, some months have more fraudulent transactions than others, suggesting that there could be months in the year where fraudsters are more likely to operate. Likewise, we can see that there’s a variation of fraudulent rate across different order days and hours of the day, showing us that certain days and hours may exhibit higher risks of fraudulent activities. Given these observations, the features such as Order Month, Order Weekday, and Order Hour can be considered as our predictors.

8.1.3 Fraudulent Rate by Late Delivery Risk

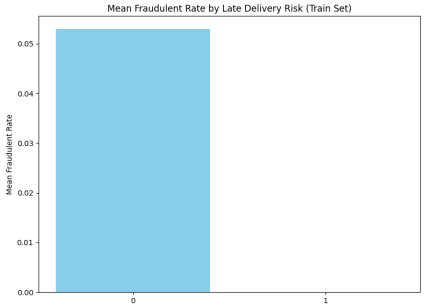

Figure 11: Bar Plot of Fraudulent Rate by Late Delivery Risk

Transactions with a late delivery risk of 0, meaning no late delivery risk, have a higher fraudulent rate nearing 0.05, while those with a risk of 1 have a near-zero fraudulent rate. Given this clear difference, transactions without a late delivery risk appear more prone to fraud in this dataset. As explained in the *EDA section*, not all orders with non late delivery risk are fraudulent, so there is no data leakage involved.

8.1.4 Fraudulent Rate by Customer Segment

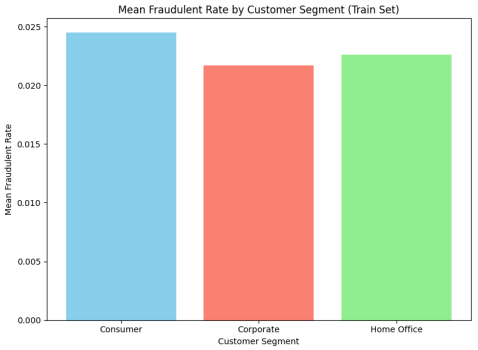

Figure 12: Bar Plot of Fraudulent Rate by Customer Segment

The bar chart shows that the average fraud rate changes depending on whether the customer is from the Consumer, Corporate, or Home Office segment. This difference tells us that knowing the customer segment can help predict fraud, as Corporate customers have a noticeably higher fraud rate at almost 0.025.

8.1.5 Fraudulent Rate by Shipping Mode

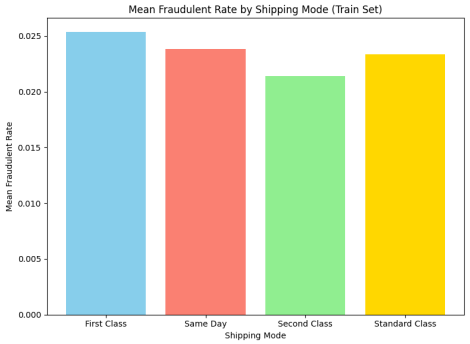

Figure 13: Bar Plot of Fraudulent Rate by Shipping Mode

As seen above, we can see there's a difference in mean fraudulent rate across the shipping modes: First Class, Same Day, Second Class, and Standard Class. The First Class segment has the highest mean fraudulent rate at slightly above 0.025, followed closely by the other shipping modes. Given the variations, shipping mode can be considered significant predictors.

8.1.6 Fraudulent Rate by Market

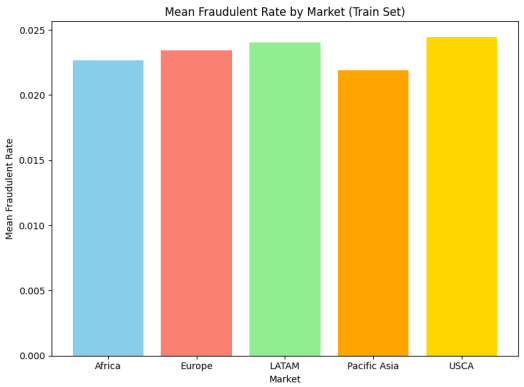

Figure 14: Bar Plot of Fraudulent Rate by Market

The given bar chart highlights that the average fraud rate varies by market, hinting that the market category might be a key factor in predicting fraudulent transactions in our supply chain order. Specifically, the USCA market records a higher average fraud rate compared to others, pointing to a greater risk of fraud in this region.

8.1.7 Fraudulent Rate by Order Country

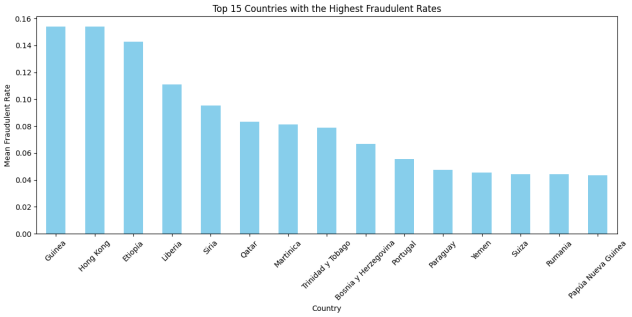

Figure 15: Bar Plot of Fraudulent Rate by Order Country

The bar chart shows that fraud rates change significantly across countries, with places like Guinea and Hong Kong showing much higher rates than others. This difference in rates by country is useful for figuring out which transactions are more likely to be fraudulent.

8.1.8 Fraudulent Rate by Department Name

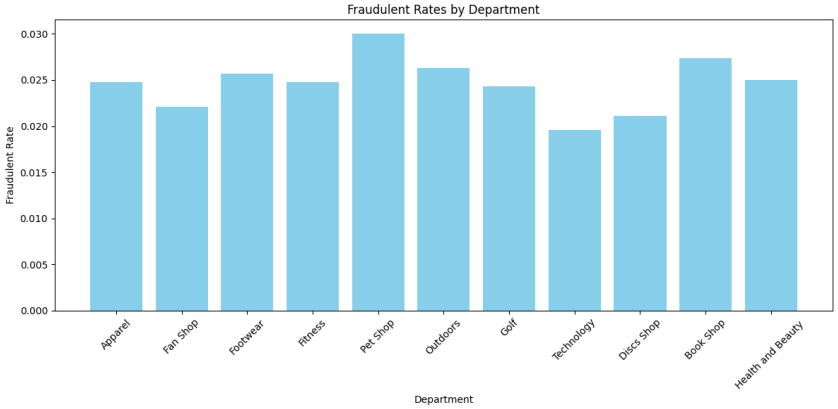

Figure 16: Bar Plot of Fraudulent Rate by Department Name

The plot shows varying fraudulent rates across different departments, showing that certain departments are more prone to fraudulent transactions. This suggests that the Department Name feature can be a significant predictor in our model, as it can help distinguish patterns of fraud specific to each department.

8.2 Feature Importances

Our group attempted to use explainable ML methods to find out the features that are the most important in the classification of fraudulent transactions. Using tree-based classifiers, we can actually plot out the feature importances. This following chart is plotted using the Gini impurity index from the Random Forest Classifier:

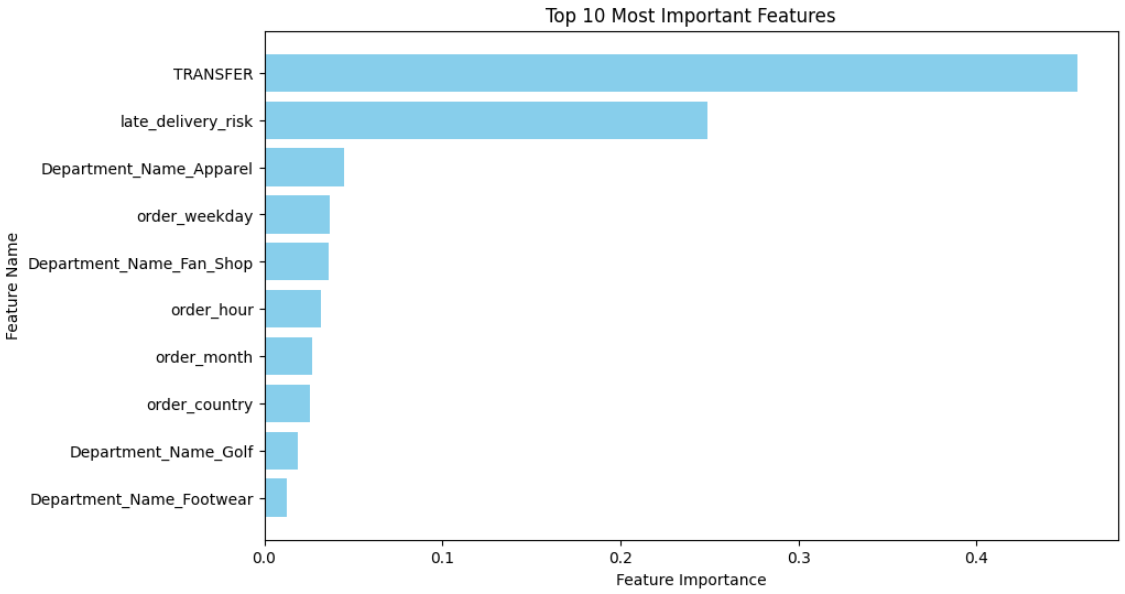

Figure 17: Feature Importance for Random Forest Classifier

Similar to the logistic regression coefficients in the report, we can see that the scenario where the TRANSFER payment method and late delivery risk are very pivotal for determining fraudulent cases. There are other factors, such as the order weekday or product department that can help classify fraudulent transactions from non-fraudulent ones.

8.3 Cross Evaluation of Train-Test Loss

This section is especially important to determine if any of our hyperparameters we are using will be causing overfitting, whereby loss is decreasing in the train-set while increasing in test-set. We have plotted a few charts of train-test loss across our hyperparameters. This will be done for the four models that we will be tuning our hyperparameters for.

8.3.1 Logistic Regression

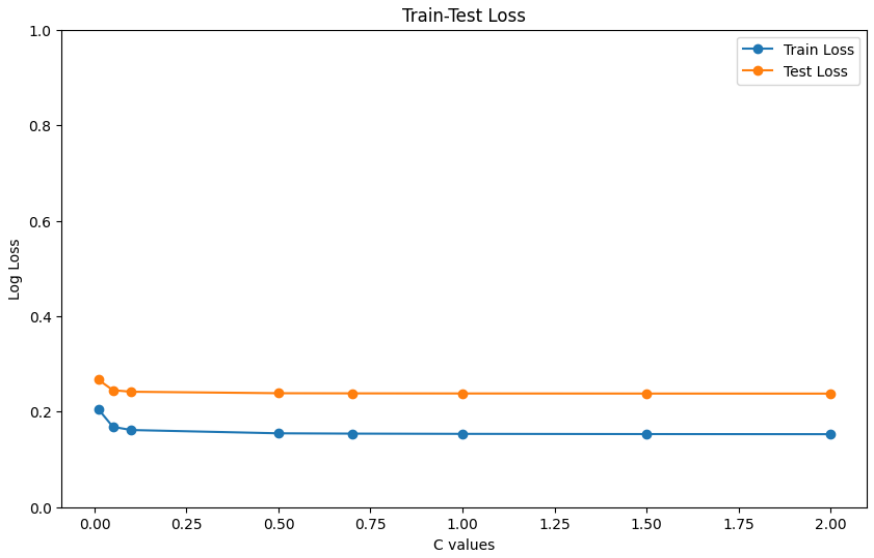

Figure 18: Train-Test Loss over C values

This is the plot of log-loss over C values. Our best model uses a C value of 0.1, and our group determines that it is a reasonable value to use for our logistic regression, with little risk of overfitting. This yielded us a great AUC score of 0.959 and recall of 1.0 in the test set. The best hyperparameters for this classifier is: {'C': 0.1, 'fit_intercept': True, 'penalty': 'l2', 'solver': 'newton-cg'}

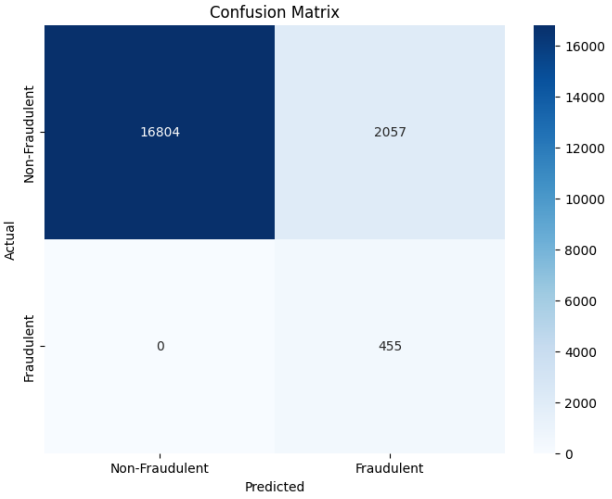

Figure 19: Confusion Matrix for Logistic Regression

8.3.2 AdaBoost

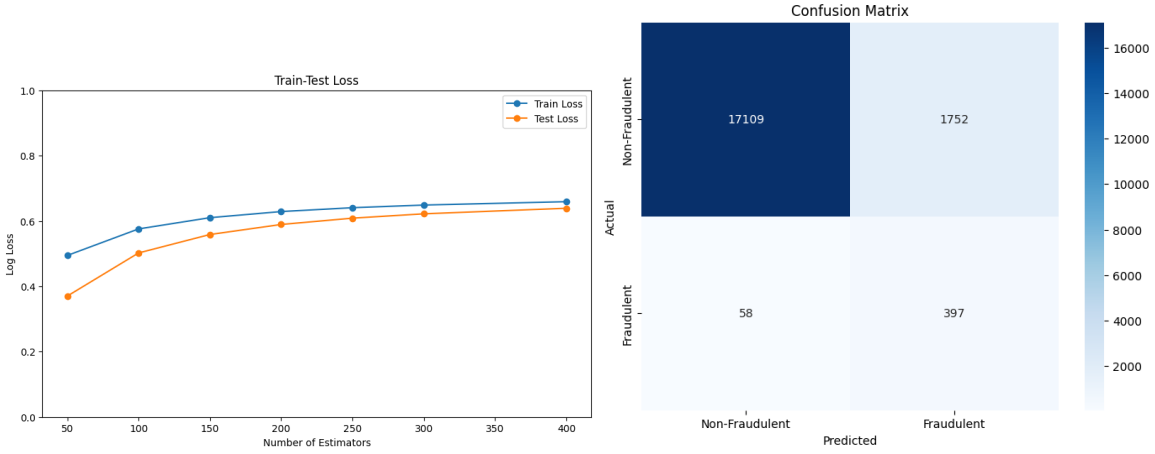

Figure 20: Plots for AdaBoost Classifier

Similarly, we determined that the hyperparameters that we are using for our AdaBoost classifier has little risk of overfitting, and it performs very well on the test-set, with an AUC of 0.957 and a recall of 0.873. The best hyperparameters for this classifier is: {'algorithm': 'SAMME.R', 'learning_rate': 1.0, 'n_estimators': 200}

8.3.3 Gradient Boosting Classifier

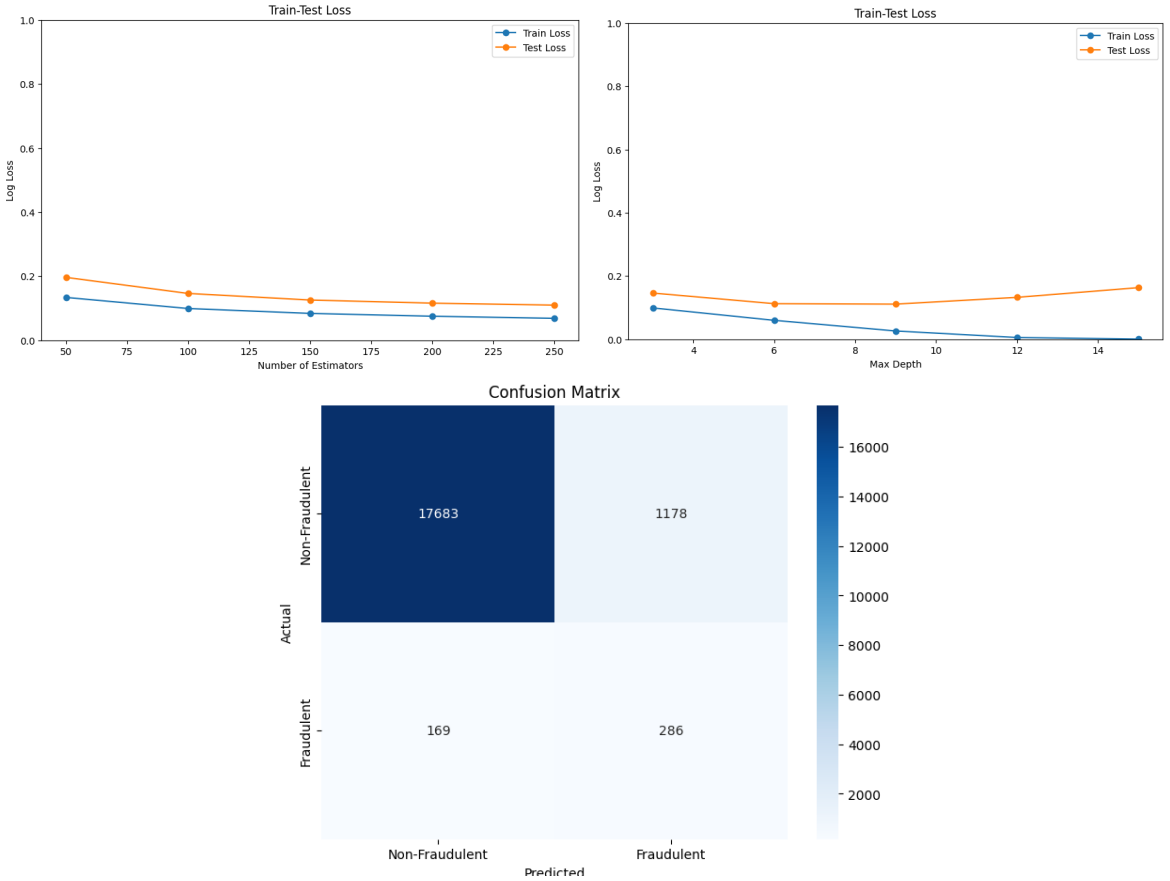

Figure 21: Plots for Gradient Boosting Classifier

We determined that the hyperparameters that we are using for our Gradient Boosting Classifier has little risk of overfitting, and it performs relatively well on the test-set, with an AUC of 0.954 but a slightly low recall of only 0.629. However, it has a slightly better precision. The best hyperparameters for this classifier is: {'criterion': 'squared_error', 'max_depth': None, 'min_samples_leaf': 3, 'min_samples_split': 4, 'n_estimators': 150, 'subsample': 0.8}

8.3.4 LightGBM Classifier

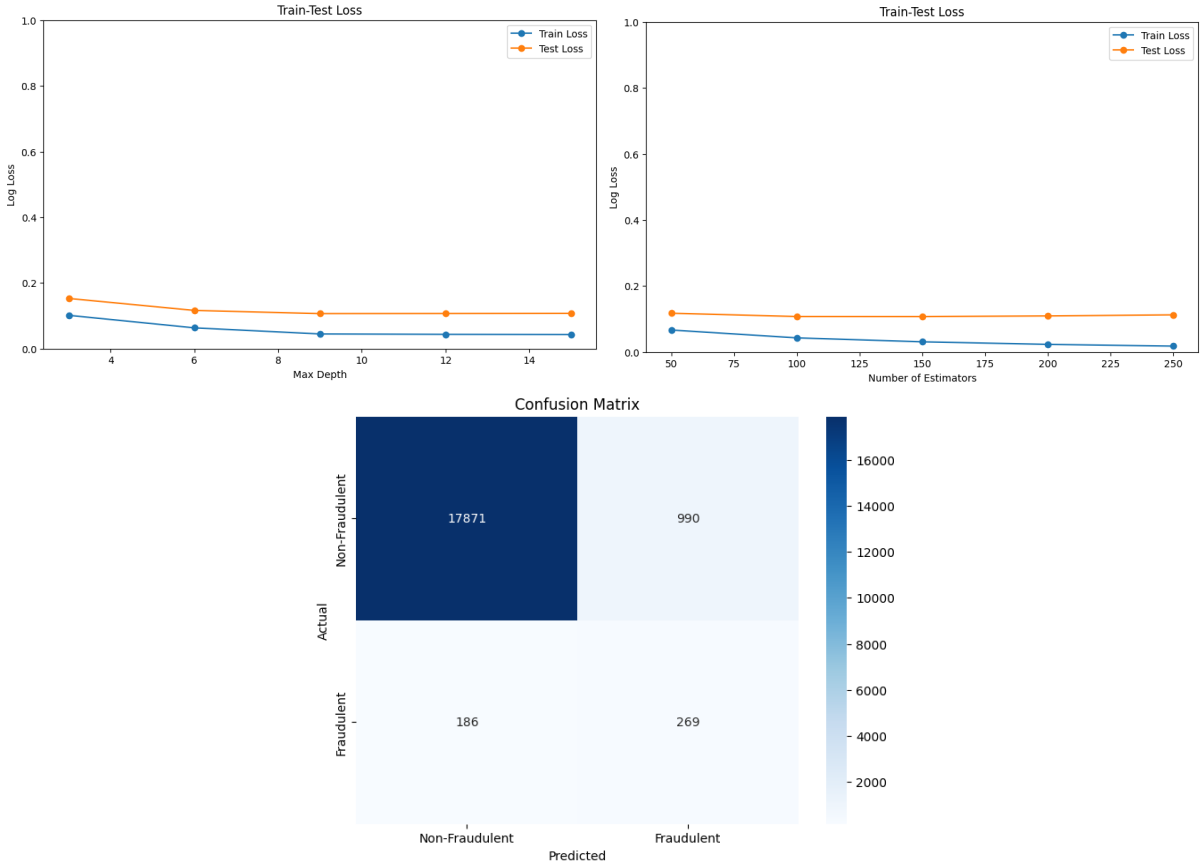

Figure 22: Plots for LightGBM Classifier

We determined that the hyperparameters that we are using for our Gradient Boosting Classifier has little risk of overfitting, and it performs relatively well on the test-set, with an AUC of 0.956 but a slightly low recall of only 0.591. The best hyperparameters for this classifier is: {'colsample_bytree': 0.8, 'max_depth': 10, 'min_child_samples': 5, 'min_child_weight': 1, 'n_estimators': 200, 'subsample': 0.6}

9. References

Abraham, W., Matthew, O., & Justin, B. (2017). Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers. *Journal of Machine Learning Research 18*. https://www.jmlr.org/papers/volume18/15-240/15-240.pdf

Alexey, N., & Alois, K. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, *7*. https://doi.org/10.3389/fnbot.2013.00021

Chen, C., Zhang, Q., Ma, Q., & Yu, B. (n.d.). Predicting protein-protein interactions through LightGBM with multi-information fusion. *Chemometrics and intelligent laboratory systems : an international journal sponsored by the Chemometrics Society*, 54-64. https://doi.org/10.1016/j.chemolab.2019.06.003

Chevalier, S. (2022, September 21). *Global retail e-commerce sales 2026*. Statista. Retrieved November 17, 2023, from https://www.statista.com/statistics/379046/worldwide-retail-e-commerce-sales/

Constante, F. (2019, March 13). *DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS [dataset]*. Mendeley. https://doi.org/10.17632/8GX2FVG2K6.5

Coppola, D. (2023, August 29). *Global e-commerce payment fraud losses 2023*. Statista. Retrieved November 17, 2023, from https://www.statista.com/statistics/1273177/ecommerce-payment-fraud-losses-globally/

Dutta, P., Suryawanshi, P., Gujarathi, P., & Dutta, A. (2019). Managing risk for e-commerce supply chains: an empirical study. *IFAC-PapersOnLine*, *52*(13), 349-354. https://doi.org/10.1016/j.ifacol.2019.11.143

Hancock, J. T., Khoshgoftaar, T. M., & Johnson, J. M. (2023, April 11). *Evaluating classifier performance with highly imbalanced Big Data - Journal of Big Data*. Journal of Big Data. Retrieved November 17, 2023, from https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00724-5

Jeni, L., Cohn, J., & Torre, F. (2013). Facing Imbalanced Data Recommendations for the Use of Performance Metrics. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4285355/

Jung, Y., & Hu, J. (2015). A K-fold Averaging Cross-validation Procedure. *Journal of Nonparametric Statistics*, *27*(2), 167-179. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5019184/

Kazi, M. A., Woodhead, S., & Gan, D. (2022). Comparing the performance of supervised machine learning algorithms when used with a manual feature selection process to detect Zeus malware. *International Journal of Grid and Utility Computing*, *13*. https://doi.org/10.1504/IJGUC.2022.126167

Kim, B. H. (2023). Development of Online Fraud Detection and Sales Prediction Model using Supply Chain Dataset. *Journal of System and Management Sciences*, *13*(2), 501-514. http://www.aasmr.org/jsms/Vol13/No.2/Vol.13.2.34.pdf

Kotsiantis, S., Kanellopoulos, D., & Pintelas, P. E. (2005, November). *Handling imbalanced datasets: A review*. ResearchGate. Retrieved November 17, 2023, from https://www.researchgate.net/publication/228084509_Handling_imbalanced_datasets_A_review

KPMG. (2017). *A holistic approach to prevention, detection, and response*. KPMG LLP. Retrieved November 17, 2023, from https://assets.kpmg.com/content/dam/kpmg/be/pdf/Markets/supply-chain-fraud.pdf

KPMG. (2020, May 7). *The supply chain fraud pandemic - KPMG Global*. KPMG International. Retrieved November 17, 2023, from https://kpmg.com/xx/en/blogs/home/posts/2020/05/supply-chain-fraud-pandemic.html

Maaten, L. v. d., & Postma, E. (2009, October 26). *Dimensionality Reduction: A Comparative Review*. Tilburg centre for Creative Computing. Retrieved November 17, 2023, from https://members.loria.fr/moberger/Enseignement/AVR/Exposes/TR_Dimensiereductie.pdf

Mishchenko, K., Khaled, A., & Richtarik, P. (2020). Random Reshuffling: Simple Analysis with Vast Improvements. *Advances in Neural Information Processing Systems*, *33*(-), -. https://proceedings.neurips.cc/paper/2020/hash/c8cc6e90ccbff44c9cee23611711cdc4-Abstract.html

Muthukrishnan, R., & Rohini, R. (2016). LASSO: A feature selection technique in predictive modeling for machine learning. https://ieeexplore.ieee.org/abstract/document/7887916

Nayak, S. K., & Ojha, A. C. (2020, March 24). Data Leakage Detection and Prevention: Review and Research Directions. *Machine Learning and Information Processing*, 203-212. https://doi.org/10.1007/978-981-15-1884-3_19

Roy, B. (2023, February 4). *All about Categorical Variable Encoding | by Baijayanta Roy*. Towards Data Science. Retrieved November 17, 2023, from https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02

Sisodia, D. S., Reddy, N. K., & Bhandari, S. (2017). Performance evaluation of class balancing techniques for credit card fraud detection. https://ieeexplore.ieee.org/abstract/document/8392219

Upadhyay, D., Manero, J., Zaman, M., & Sampalli, S. (2021, March). Gradient Boosting Feature Selection With Machine Learning Classifiers for Intrusion Detection on Power Grids. *IEEE Transactions on Network and Service Management*, *18*, 1104-1116. 10.1109/TNSM.2020.3032618

Wang, C., Sun, D., & Toh, K. (2010). Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm. https://doi.org/10.1137/090772514

Zhang, S., Cheng, D., & Hu, R. (2018). Supervised feature selection algorithm via discriminative ridge regression. 1545–1562. https://doi.org/10.1007/s11280-017-0502-9

Zhang, S., Wu, J., & Jia, Y. (2021, April). A temporal LASSO regression model for the emergency forecasting of the suspended sediment concentrations in coastal oceans: Accuracy and interpretability. *Engineering Applications of Artificial Intelligence*, *100*. https://doi.org/10.1016/j.engappai.2021.104206

Zhou, Y., Li, H., Xiao, Z., & Qiu, J. (2023). A user-centered explainable artificial intelligence approach for financial fraud detection. *Finance Research Letters*, *58*(A), -. https://www.sciencedirect.com/science/article/abs/pii/S1544612323006815