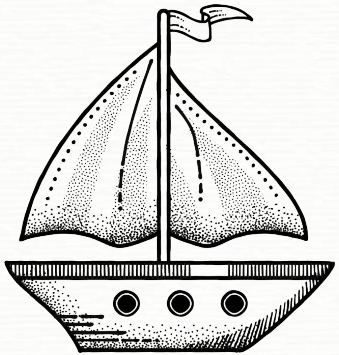
Fraudulent Prediction on E-commerce transactions

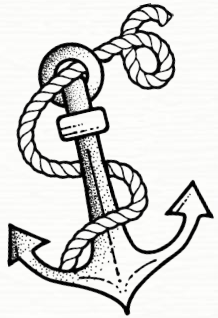
Presented by The Mugiwara Pirates
With Adam, Armen, Kranta & Stéphanie

01

Introduction

Project overview





Dataset

E-commerce transactions

Goal

Predict fraudulent transactions

Questions

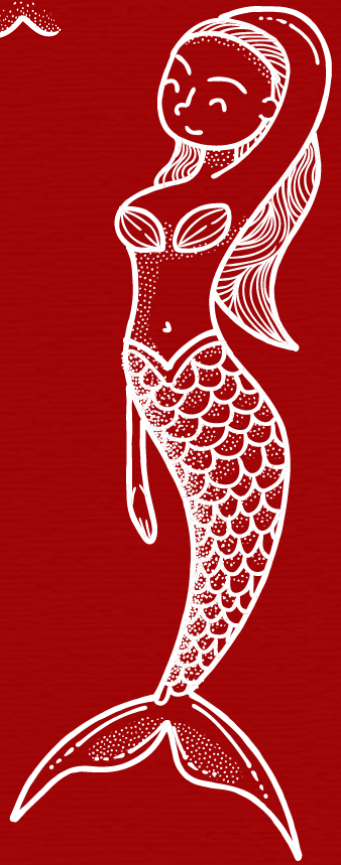
Can we identify distinct profiles or characteristics of customers who are more likely to be involved in fraudulent transactions ?

What are the most important features or characteristics that contribute to the likelihood of a transaction being fraudulent ?

Can we build an accurate machine learning model to predict whether a transaction is fraudulent or not ?

02

Data selection and preparation



1,491,586 rows

Total number before sampling

75,060

Fraudulent data

150,000

After undersampling (50/50)

Data cleaning

Null values

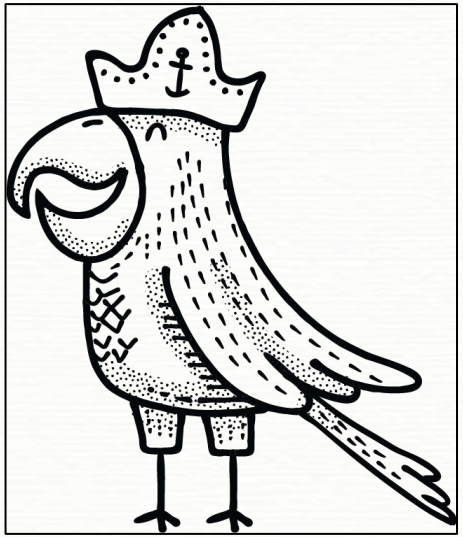
No null values to remove

Duplicates

No duplicates

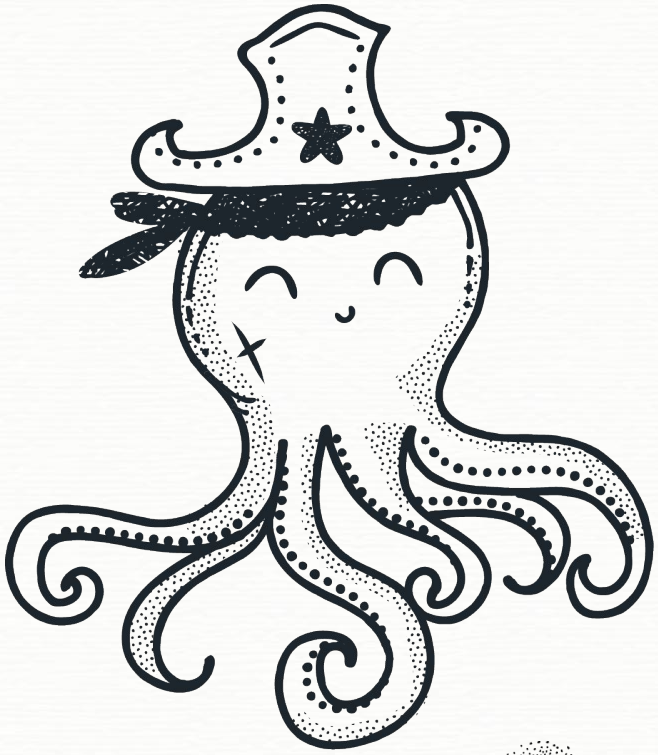
Wrong data

Negative age values → replaced by median age



03

Feature engineering and selection



Features

- Transaction ID
- Customer ID
- Transaction Amount
- Transaction Date
- Payment Method
- Product Category
- Quantity
- Customer Age
- Customer Location
- Device Used
- IP Address
- Shipping Address
- Billing Address
- Is Fraudulent
- Account Age Days
- Transaction Hour

Features

- ~~Transaction ID~~
- ~~Customer ID~~
- Transaction Amount
- ~~Transaction Date~~
- Payment Method
- Product Category
- Quantity
- Customer Age
- ~~Customer Location~~
- Device Used
- ~~IP Address~~
- ~~Shipping Address~~
- ~~Billing Address~~
- Is Fraudulent
- Account Age Days
- Transaction Hour

Feature Selection

Numerical

- TRANSACTION AMOUNT
- QUANTITY
- CUSTOMER AGE
- ACCOUNT AGE DAYS
- TRANSACTION HOUR

Added:
- DAY OF THE WEEK
- MONTH

Scaling: MINMAX

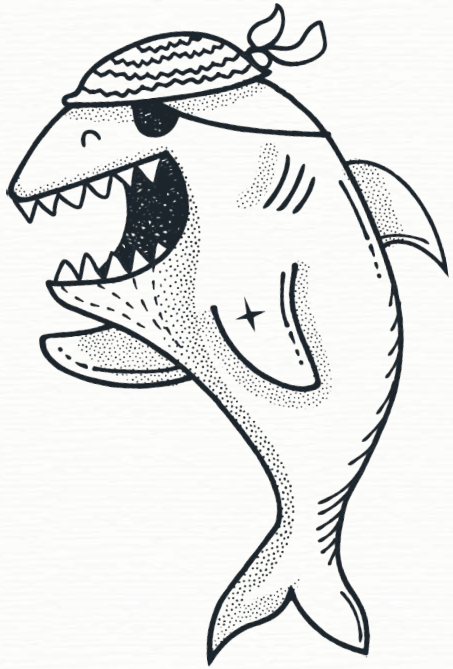
Categorical

- PAYMENT METHOD
- PRODUCT CATEGORY
- DEVICE USED
- ADDRESS MATCH

Action taken:
ONE-HOT ENCODING

05

Model Building and evaluation



Models tested

Classifier Model	Precision		Recall		F1		Accuracy
	0	1	0	1	0	1	
Random Forest	0.73	0.76	0.79	0.70	0.76	0.73	0.74
Ada boost	0.72	0.76	0.79	0.69	0.75	0.72	0.74
Bagging	0.73	0.77	0.79	0.70	0.76	0.73	0.75
Bagging Bootstrap	0.73	0.76	0.78	0.71	0.76	0.73	0.75
Gradient boosting	0.72	0.75	0.78	0.69	0.75	0.72	0.73

Models tested

Classifier Model	Precision		Recall		F1		Accuracy
	0	1	0	1	0	1	
Random Forest	0.73	0.76	0.79	0.70	0.76	0.73	0.74
Ada boost	0.72	0.76	0.79	0.69	0.75	0.72	0.74
Bagging	0.73	0.77	0.79	0.70	0.76	0.73	0.75
Bagging Bootstrap	0.73	0.76	0.78	0.71	0.76	0.73	0.75
Gradient boosting	0.72	0.75	0.78	0.69	0.75	0.72	0.73

Models tested

Classifier Model	Precision		Recall		F1		Accuracy
	0	1	0	1	0	1	
Random Forest	0.73	0.76	0.79	0.70	0.76	0.73	0.74
Ada boost	0.72	0.76	0.79	0.69	0.75	0.72	0.74
Bagging	0.73	0.77	0.79	0.70	0.76	0.73	0.75
Bagging Bootstrap	0.73	0.76	0.78	0.71	0.76	0.73	0.75
Gradient boosting	0.72	0.75	0.78	0.69	0.75	0.72	0.73

06

Hyperparameter tuning

Grid Search Cross Valuation

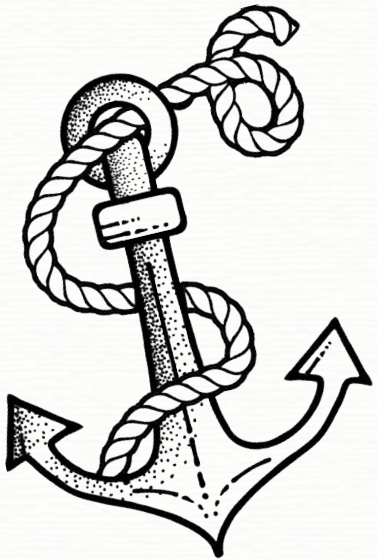
```
grid = {"n_estimators": [50, 100,500],
        "estimator__max_depth":[10, 40, 100],
        'max_samples': [0.5, 0.75, 1.0],}

model = GridSearchCV(estimator = bagging_class, param_grid = grid, cv=5)
model.fit(X_train_norm, y_train)
```

✓ 595m 42.4s

	precision	recall	f1-score	support
0	0.73	0.78	0.76	15169
1	0.76	0.71	0.73	14831
accuracy			0.75	30000
macro avg	0.75	0.75	0.75	30000
weighted avg	0.75	0.75	0.75	30000

07

Key findings



Thanks!

Do you have any questions?