Project Report

Predicting fraudulent credit card transactions

Data Science for Business I

Group 31

Business Problem and Project Goals

Credit card fraud is a serious business problem. Despite their fighting efforts, the card payment industry faces billions in losses annually due to fraudulent credit card transactions. Nilson Report from 2021 finds that in 2020 credit card issuers, merchants and acquirers of merchant and ATM transactions collectively lost $28.58 billion to card fraud. The report estimates that the industry will face $408.50 billion in losses over the next decade globally.

Global fraud and payments report from 2022 by CyberSource states that most of the credit card fraud happens online and that merchants spend an average of 10% of their eCommerce revenues to manage payment fraud. The report finds that fraud management is regarded as challenging by 91% of merchants and that some of the top fraud detection challenges are identifying and responding to emerging fraud attacks, effectively using data to manage fraud and updating fraud risk models.

Even though card issuers and merchants are already combating fraud with several techniques and some predictive models are already in use, due to the severity of the issue, we wanted to create our own predictive model to efficiently and accurately detect fraudulent transactions. **The goal of our project is to make it faster and easier to detect credit card fraud and that way create savings for credit card issuers and merchants** and benefit customers in form of safer use of cards.

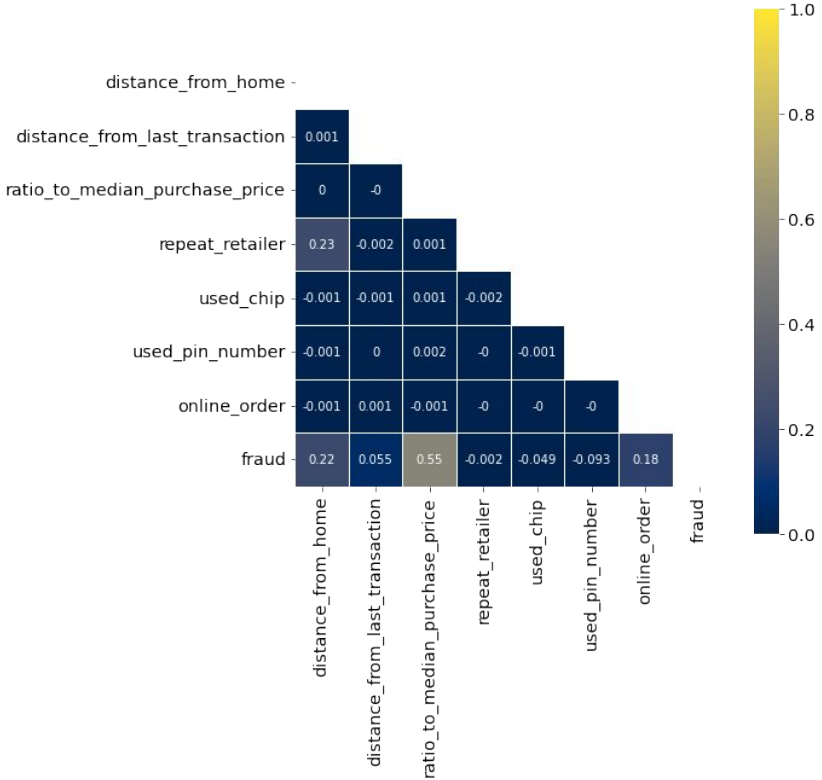
Data used for our project

The dataset used for this project is from Kaggle.com and it consists of 1 000 000 rows and 8 variables. No variable column in the dataset has null values.

Variables

Name	Description	Type	Values
fraud	1 if the transaction is fraudulent, 0 otherwise	binary	0/1
distance_from_home	Distance between where transaction happened and home	continuous	0.005-10632.724
distance_from_last_transaction	Distance between current and last transaction	continuous	5.037-11851.105
ratio_to_median_purchase_price	Transaction amount / median purchase	continuous	0.004-267.803
repeat_retailer	1 if transaction happened at a previous retailer	binary	0/1
used_chip	1 if chip was used	binary	0/1
used_pin_number	1 if pin was used	binary	0/1
online_order	1 if the order was done online	binary	0/1

Correlation matrix of the variables



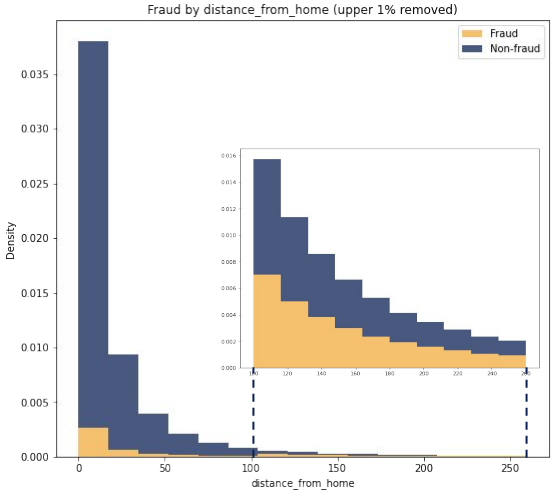
A few notably **high** correlations

‘Fraud’ & ‘Ratio_from_purchase_price’	**0.55**
‘Fraud’ & ‘Distance_from_home’	0.22
‘Fraud’ & ‘Online_order’	0.18
‘Repeat_retailer’ & ‘Distance_from_home’	0.23

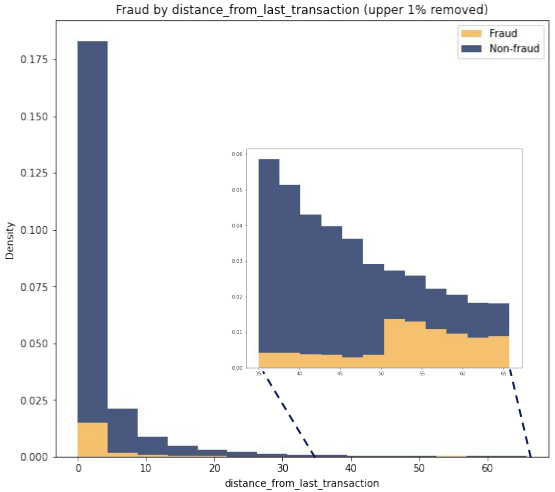
Density of fraud transactions by continuous variables

The data is very unbalanced - there are significantly more non-fraudulent data points than fraudulent ones

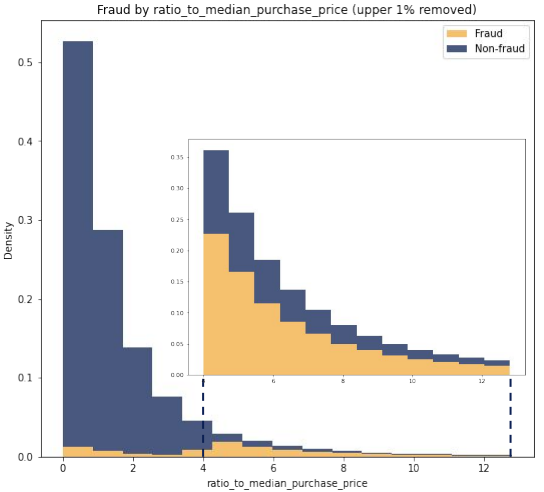
- 87403 fraudulent transactions
- 8,74 % of all transactions



Possibly **slightly less fraud** when **closer** to **home, but increases** when **distance greater** than around **100**

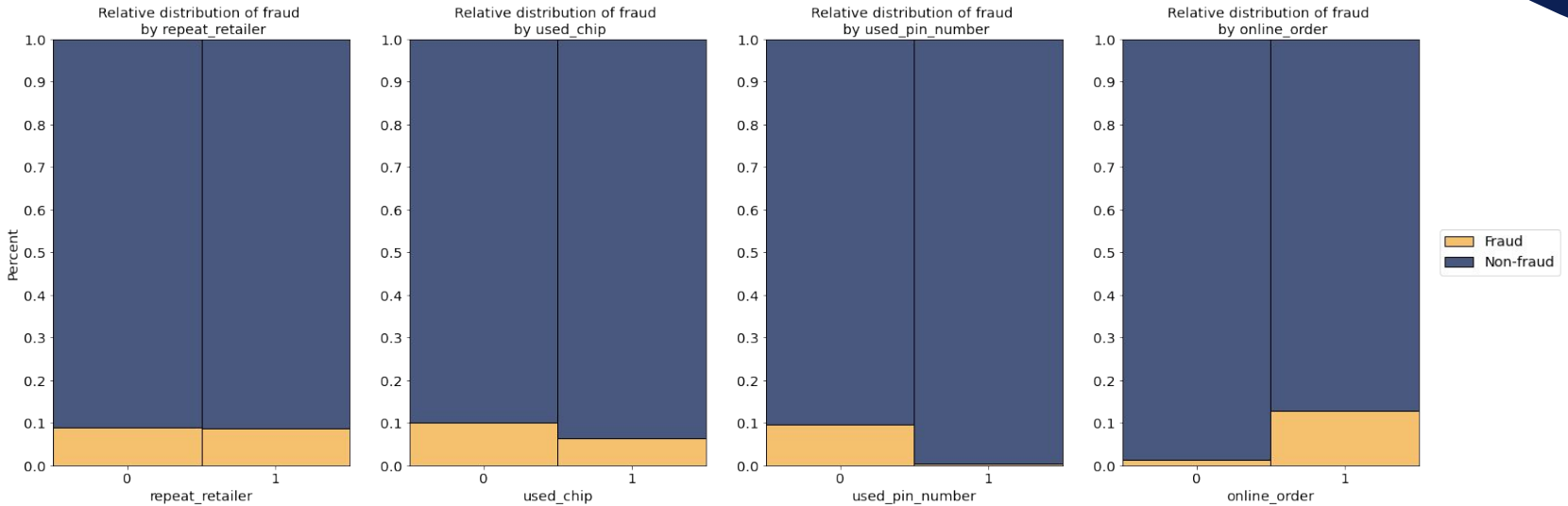


Possibly **slightly less fraud** when the **last transaction** is **closer**, **but increased significantly** when distance **greater** than around **50**



Transactions that have **ratio greater** than **4** have **remarkably higher** chance of being **fraudulent**

Relative distribution of fraud by binary features



- **Fraudulent** transactions **decreased** by a few when **chip** is **used**.
- **Fraudulent** transactions occurred **mostly** when they are **online orders** or **not authorized** by using **pin number**

Preprocessing

Data modification

- Our dataset has the three continuous variables, distance_from_home, distance_from_last_transaction and ratio_to_median_purchase_price. All of these continuous variables contained outliers above the 99th quantile, so we decided to remove the upper 1%.
- When plotting the relative distribution and the correlation matrix, we found out that the variable `repeat_retailer` does not have any effect on predicting a fraudulent case and therefore we left that variable out of our model.
- Our dataset did not include any null values, so we did not have to remove any values.

Preparation for modeling

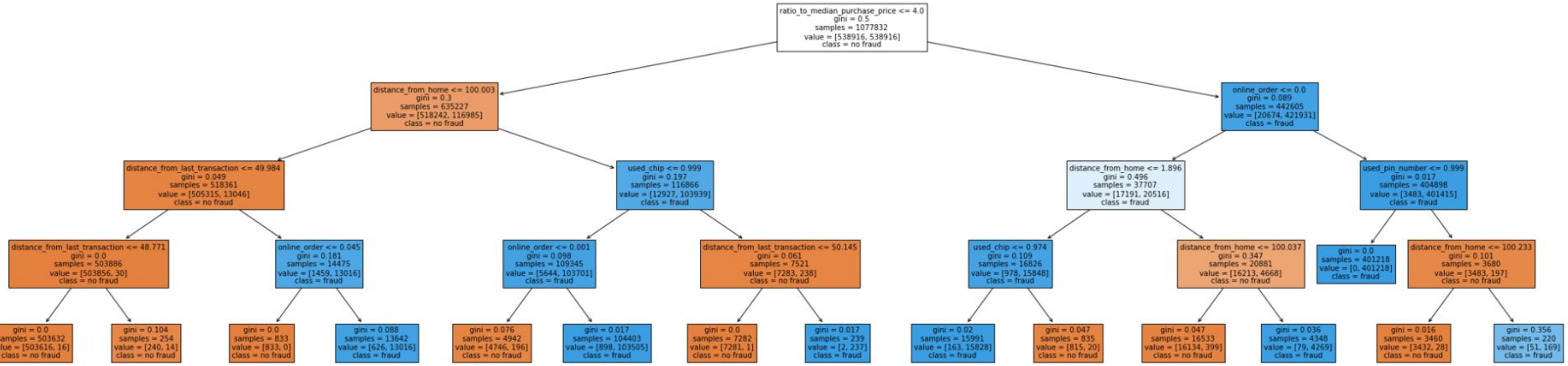
- After modifications we had a dataset with 970,291 transactions with 72,165 fraudulent cases (7.44%) for building models.
- We split the data into 3 sets: training (60%), validating (20%), and testing (20%).
- Because the data was imbalance, the minority class in the training set was resampled using SMOTE.
- The features used for logistic regression were scaled to fit from 0 to 1 to make it intuitive to interpret the regression coefficients (features were not scaled for the decision tree model)

Logistic regression

Feature	Coefficient
distance_from_home	11.81
distance_from_last_transaction	5.66
ratio_to_median_purchase_price	**21.95**
used_chip	-1.36
used_pin_number	**-12.41**
online_order	5.12

- The findings are in line with the previous slides
- Only two variables have a negative coefficient: used_chip and used_pin_number
  - This indicates that using a chip or pin_number is associated with a reduction in the relative risk of fraud
- The opposite is true for the positive coefficient for example, fraud is associated with a longer distance from home

Decision tree



Comparing models

Since the label is binary with values of 0/1, we decided to compare two models known to suit classification problems: logistic regression and a decision tree model. The training data was balanced using SMOTE due to the highly imbalance nature of the data.

	Precision	Recall	F1-Score
0.0	1.00	0.94	0.97
1.0	0.55	0.96	0.70
Accuracy			0.94

Logistic Regression

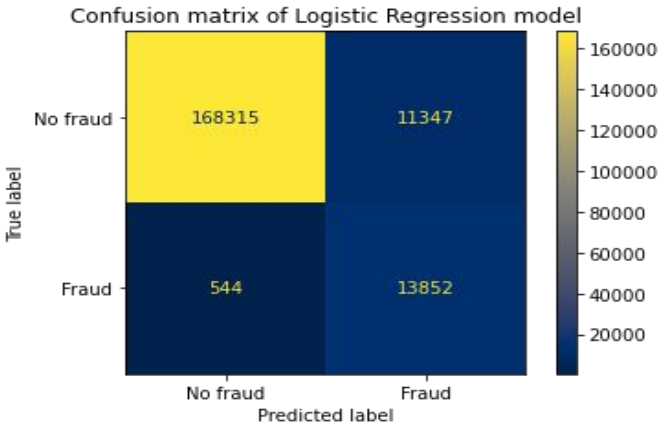
	Precision	Recall	F1-Score
0.0	1.00	1.00	1.00
1.0	0.96	1.00	0.98
Accuracy			1.00

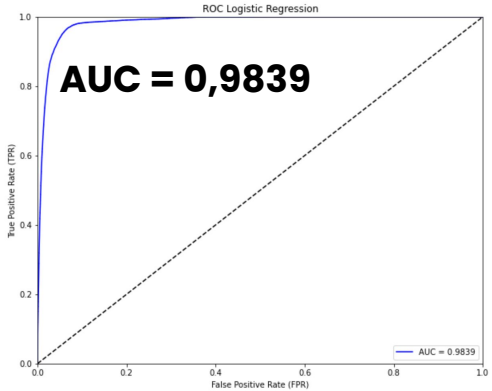
Decision Tree

- Both models have very high accuracy but as the data is very imbalance we can't solely trust this
- All precision and recall values are higher for the decision tree model but **importantly the recall for the fraud class is perfect**. We are particularly interested in this because it is more expensive to incorrectly classify fraud as non-fraudulent (false negative) but relatively inexpensive to classify a non-fraud transaction as fraud (false positive).

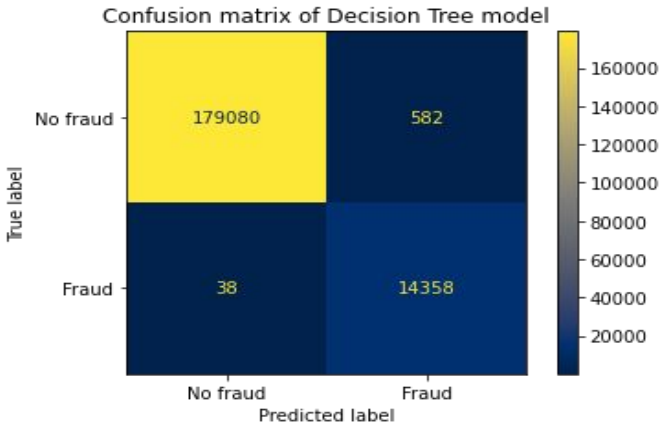
Confusion matrices and ROC curves

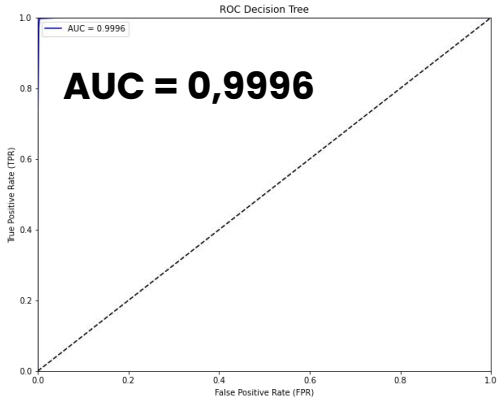
Logistic Regression





Decision Tree





Decision Tree has significantly less false negatives (and false positives).

This means more savings because false negatives are expensive

Evaluation

Finally we analysed the chosen model (Dtree) using the test data. Metrics for evaluation are on the right. The model performs significantly better than a random model according to the ROC curve.

Precision/Recall/Accuracy were at the same level as for the validation data.

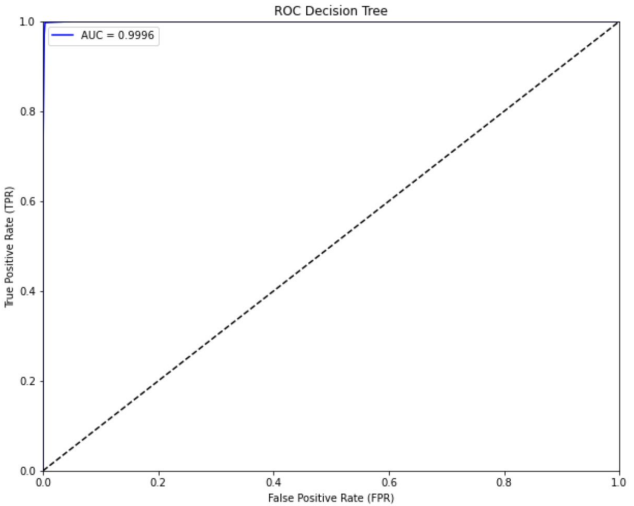
The success of the model (extremely high AUC etc) could be due to original data being simulated (real life banking data almost impossible to get due to privacy/security concerns) and our models approximating the method used to create the data very closely.

Overall we recommend the bank/company whose data this (hypothetically) is to use the model to predict fraud.

Based on our findings banks could also implement the following steps to help curb fraud:

- Always require a pin number for large purchases and purchases away from usual transactions
- Ask clients to inform that they are abroad/temporarily close a card if it is used abroad without notification

	Precision	**Recall**
0.0	1.00	1.00
1.0	0.96	1.00



References

CyberSource, Global fraud and payments report 2022
https://www.cybersource.com/content/dam/documents/campaign/fraud-report/global-fraud-report-2022.pdf

Nilson Report 2021, ISSUE 1209
https://nilsonreport.com/upload/content_promo/NilsonReport_Issue1209.pdf

Dataset:
https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud