Detecting credit card fraud

James Hodgson – GA Data Science Immersive – Capstone Project

Contents

Background & aims

- Project aim
  - To build a credit card fraud classifier to identify fraudulent transactions based on the characteristics of the transaction, the customer and their credit card.
- What is credit card fraud?
  - When your credit card details are stolen and used by fraudsters to make purchases.
  - Accounts for a very small proportion of all transactions.
  - Impacts the customer and the credit-card company.
- Success measures
  - Classifier performance, particularly accuracy.
  - Business key-performance indicators:
    - What proportion of fraudulent transactions do we detect?
    - What proportion of all transactions do we incorrectly classify as fraud?

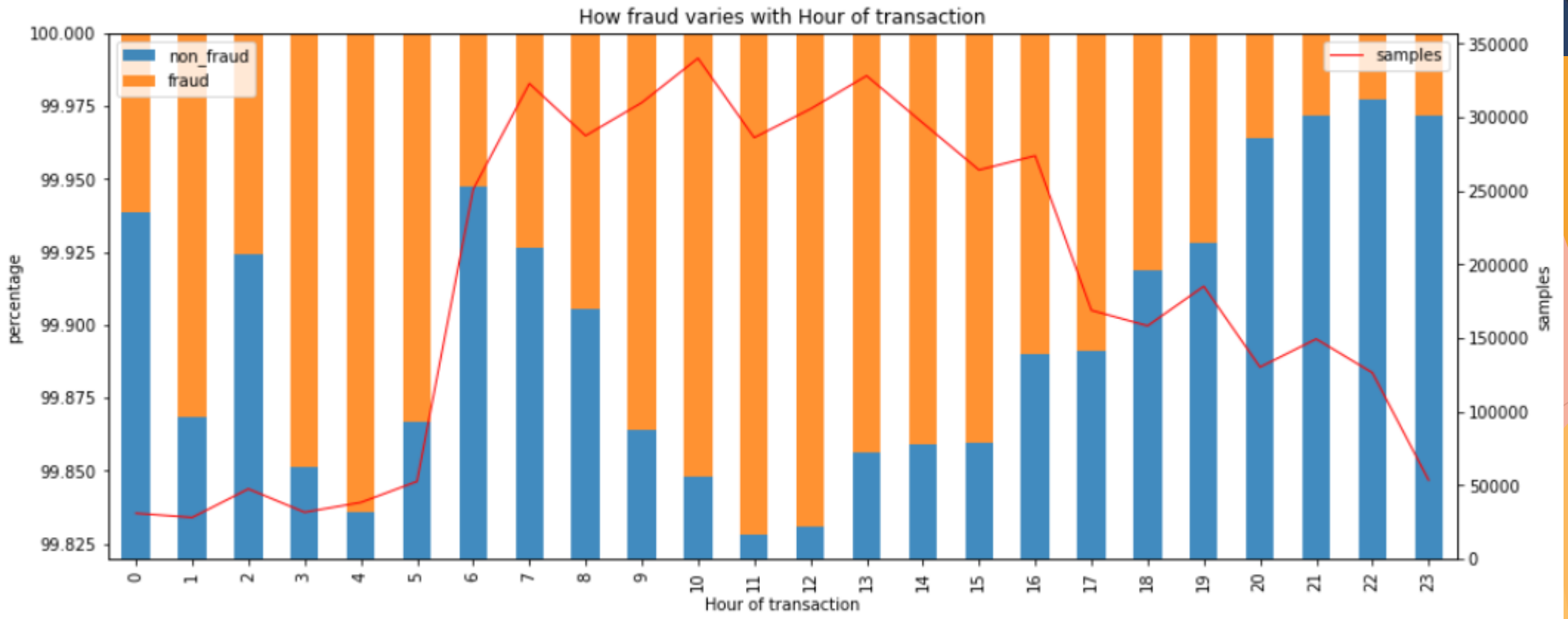
The data

The data

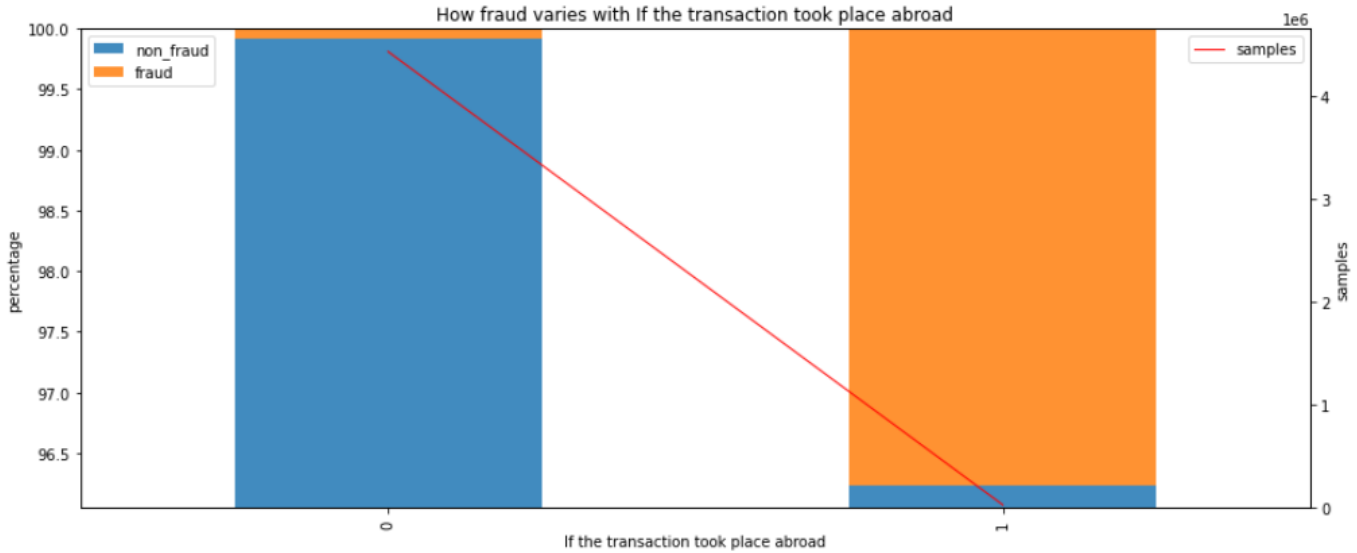
- Simulated by IBM and on Kaggle here: https://www.kaggle.com/datasets/ealtman2019/credit-card-transactions
- 24M+ credit card transactions going back 25 years.
  - Date & time, value, merchant details, location, and target
- 2000 users, based in the USA
  - Name, address, demographics, credit score
- 6000+ credit cards
  - Card type, features, details, brand
- Above joined together.
- c50 features per transaction
- **Massive class imbalance** – 99.88% of transactions not fraud.

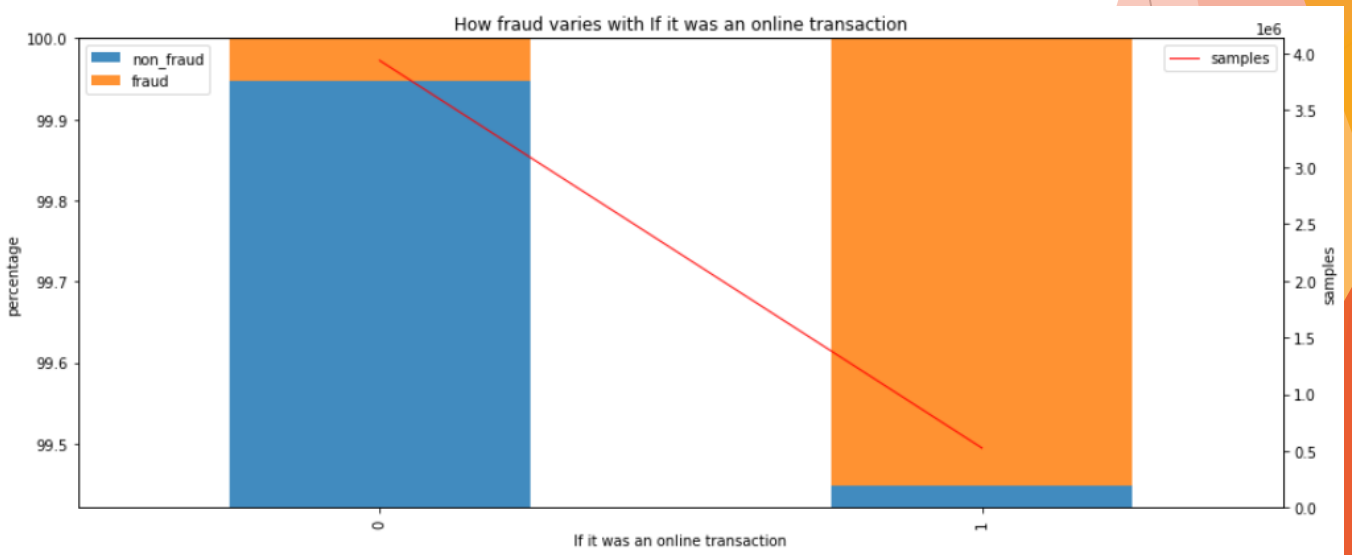
EDA – who is a fraudster and who is his prey?

The fraudster likes his evenings off but does the school run



He likes remote work



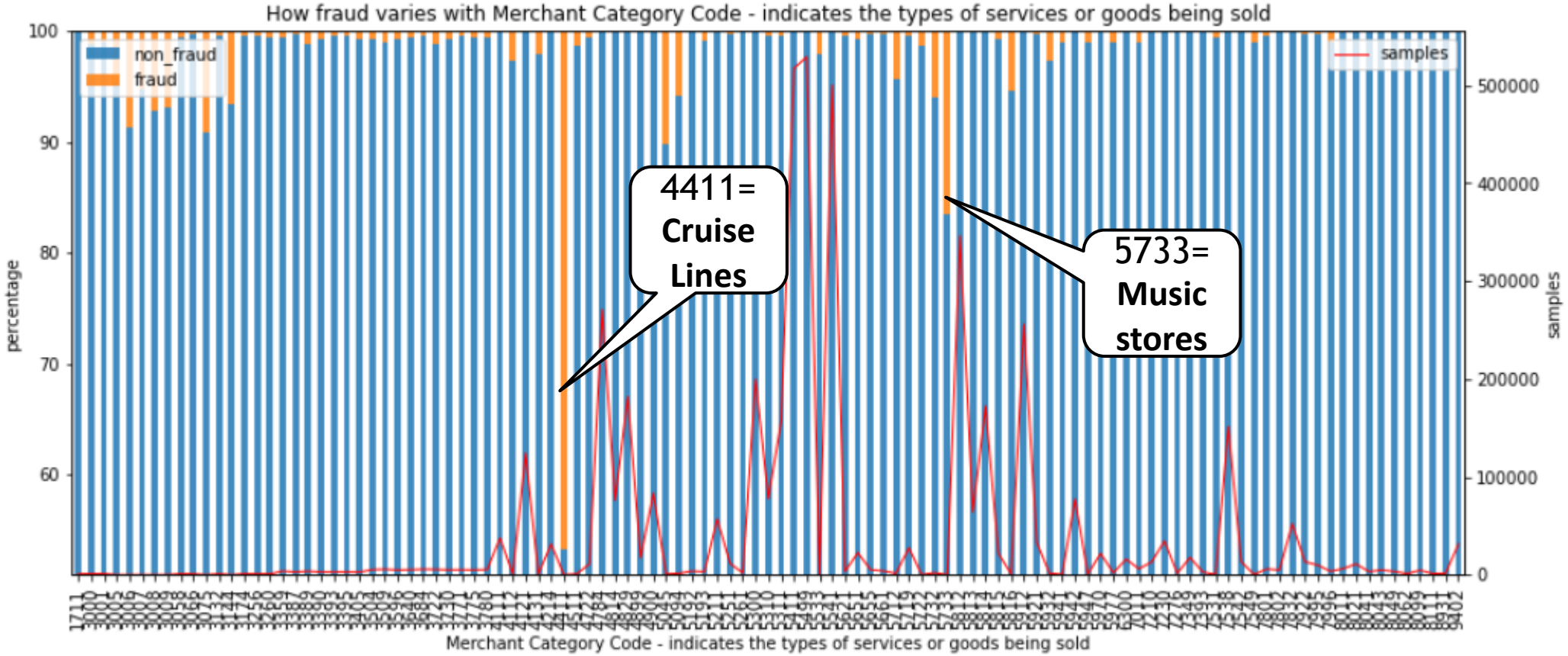


Where the fraud happens (red) and who it happens to (blue)

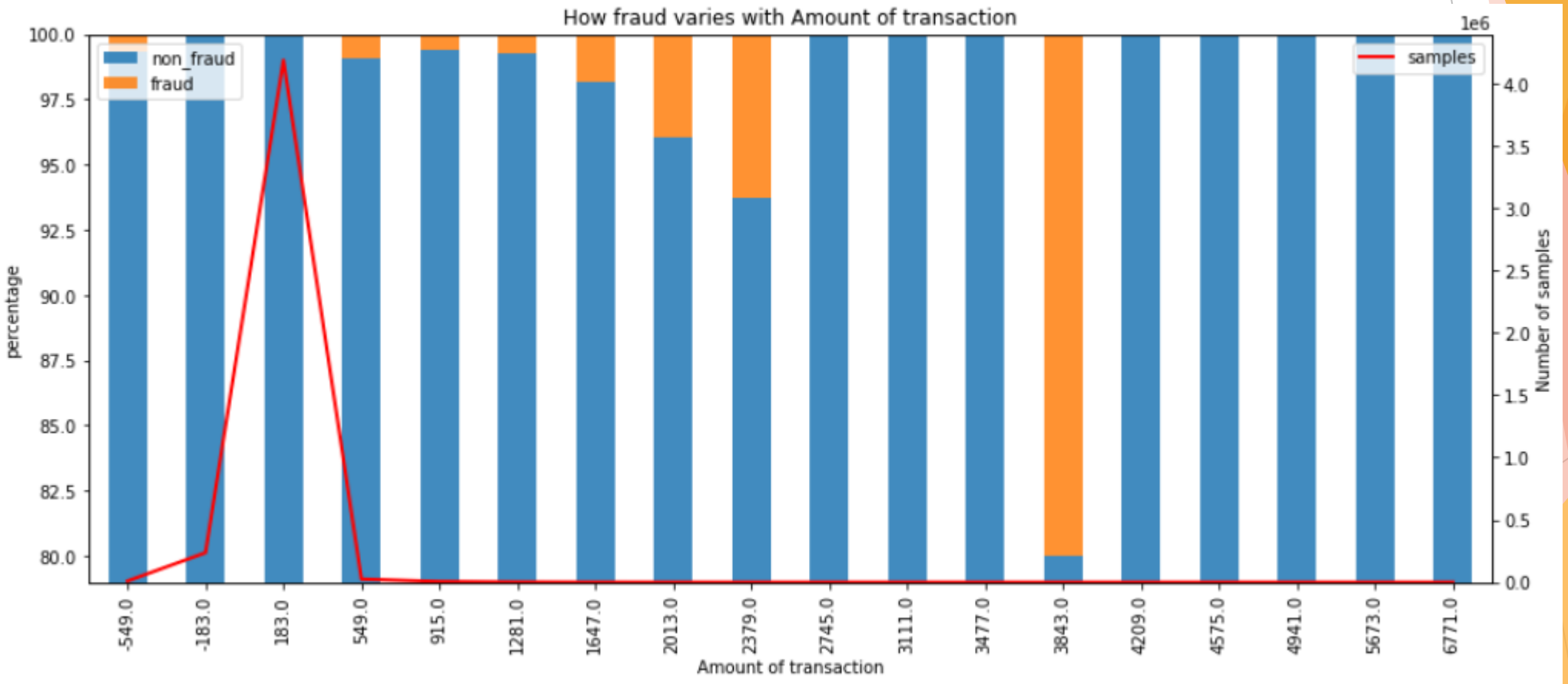
Map based on Longitude (generated)andLongitude (generated) and Latitude (generated). For pane Longitude (generated): Details are shown for Tx Zip. For pane Longitude (generated) (2): Details are shown for Usr Zipcode.The data is filtered on Tx Is Fraud, which includes values greater than or equal to 1.

.. But in-person fraud is local

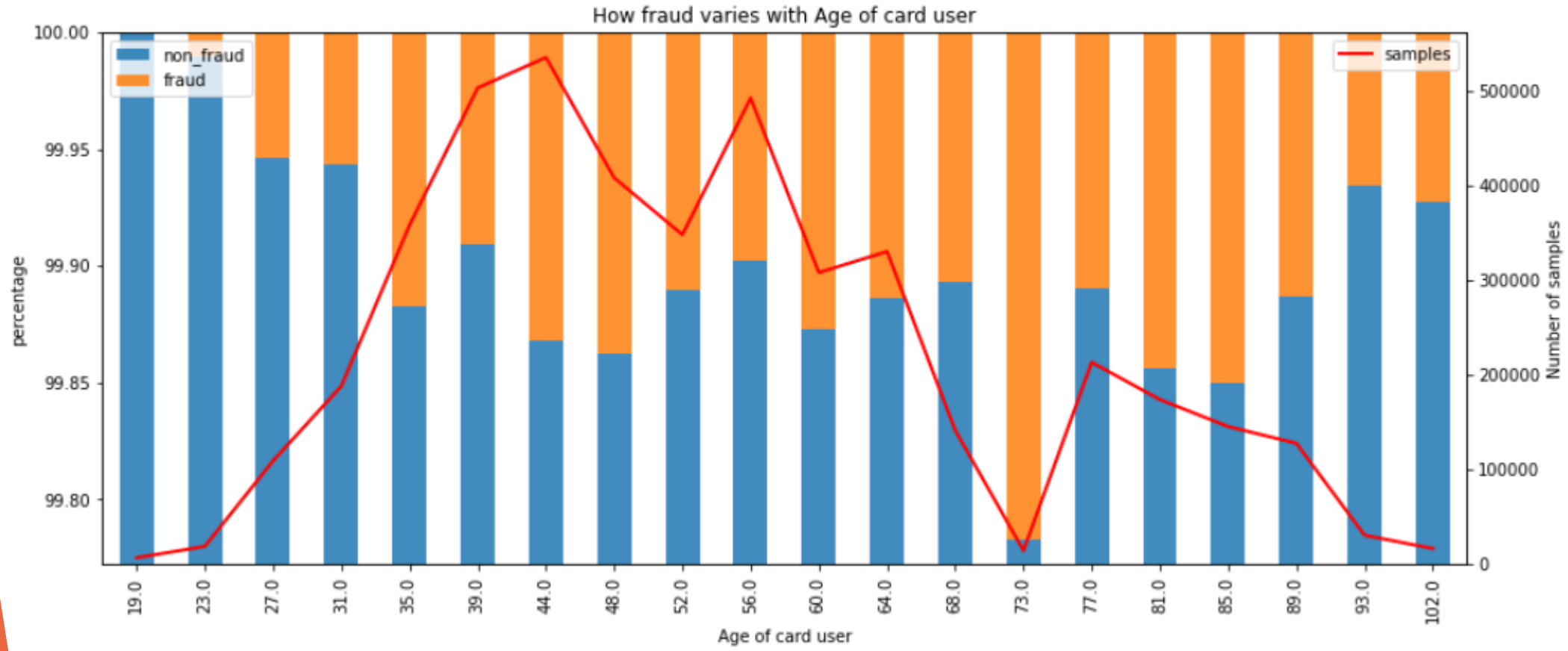
He likes cruises on the med and music



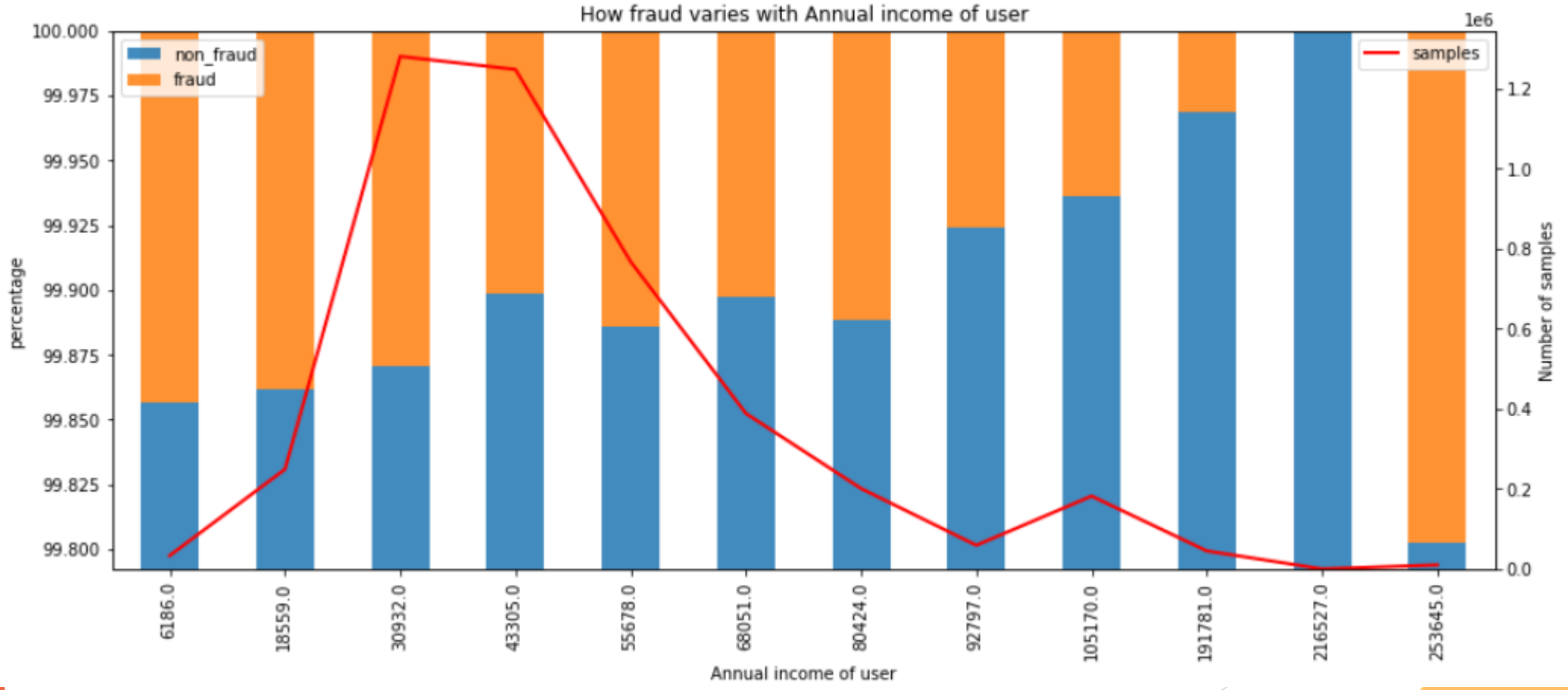
He likes to “go large”



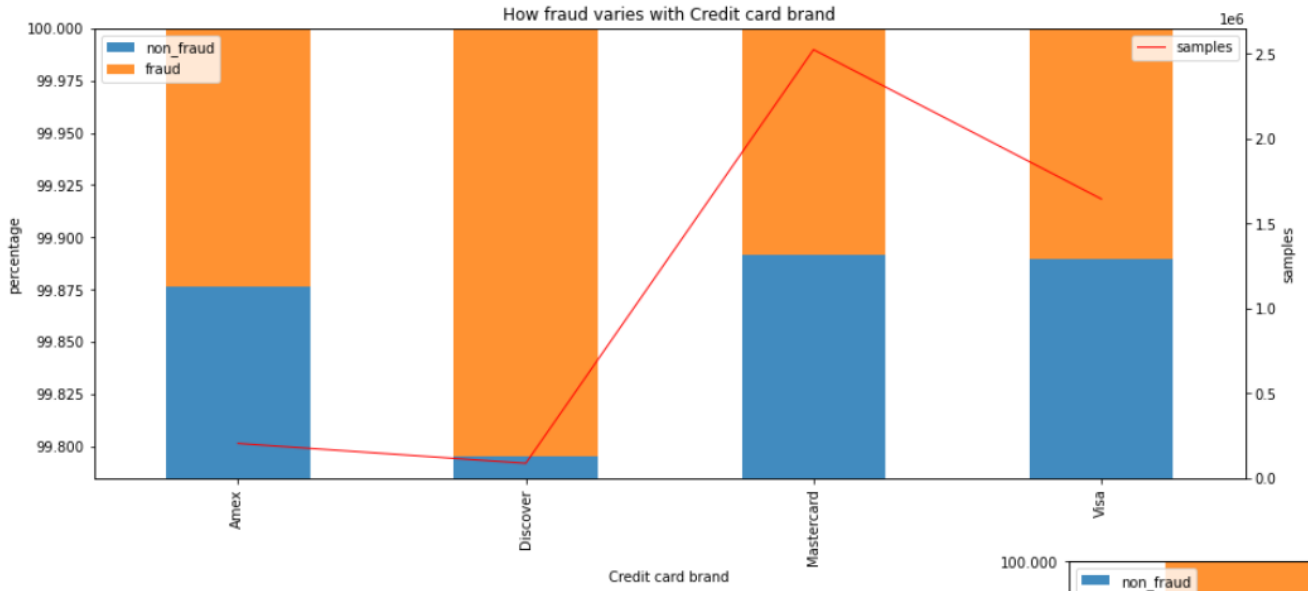
He preys on the elder …

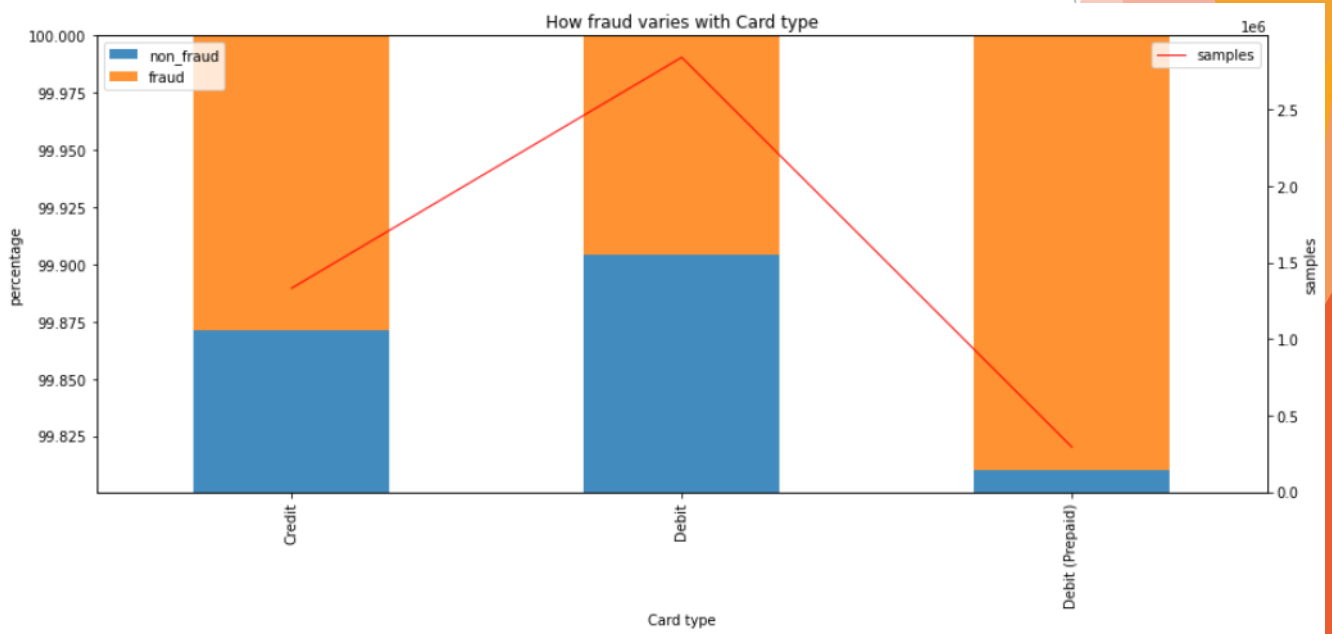


.. the less well off ...



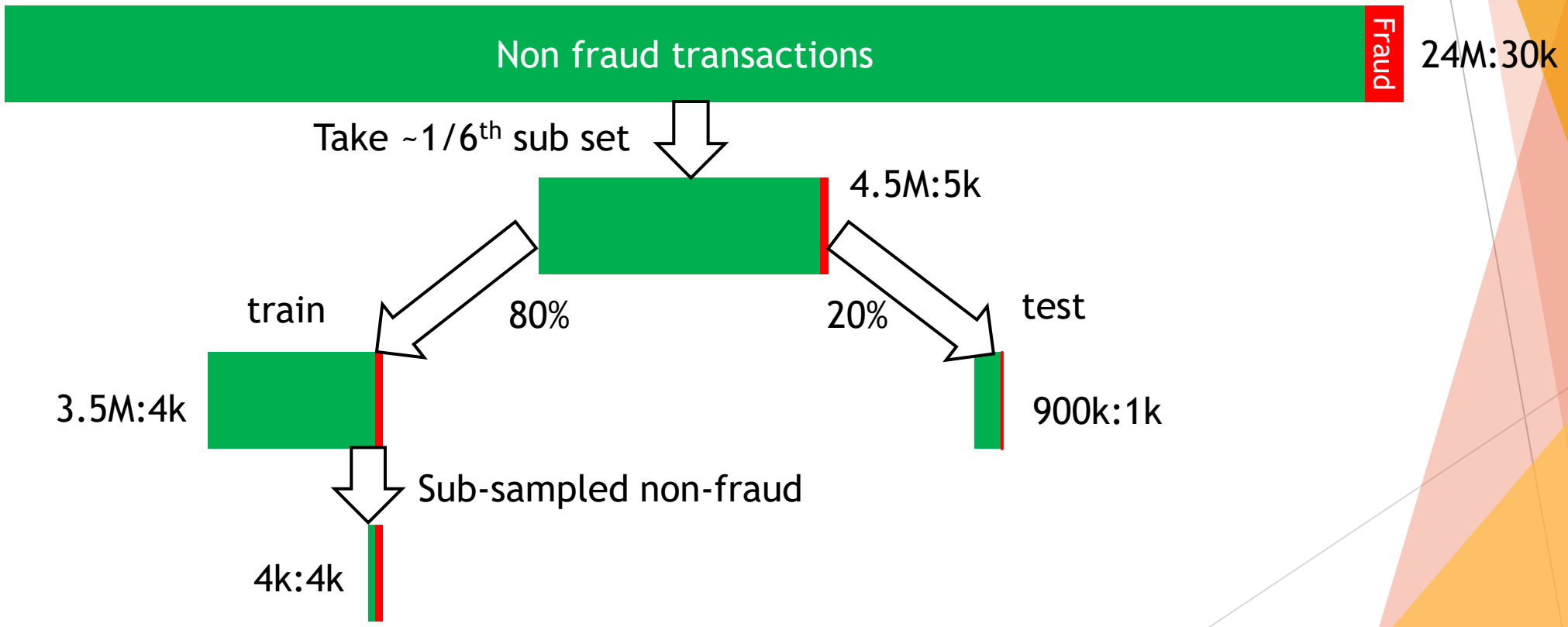
… and Discover Debit Cards





Building the fraud detection classifier

Managing the data



Modelling

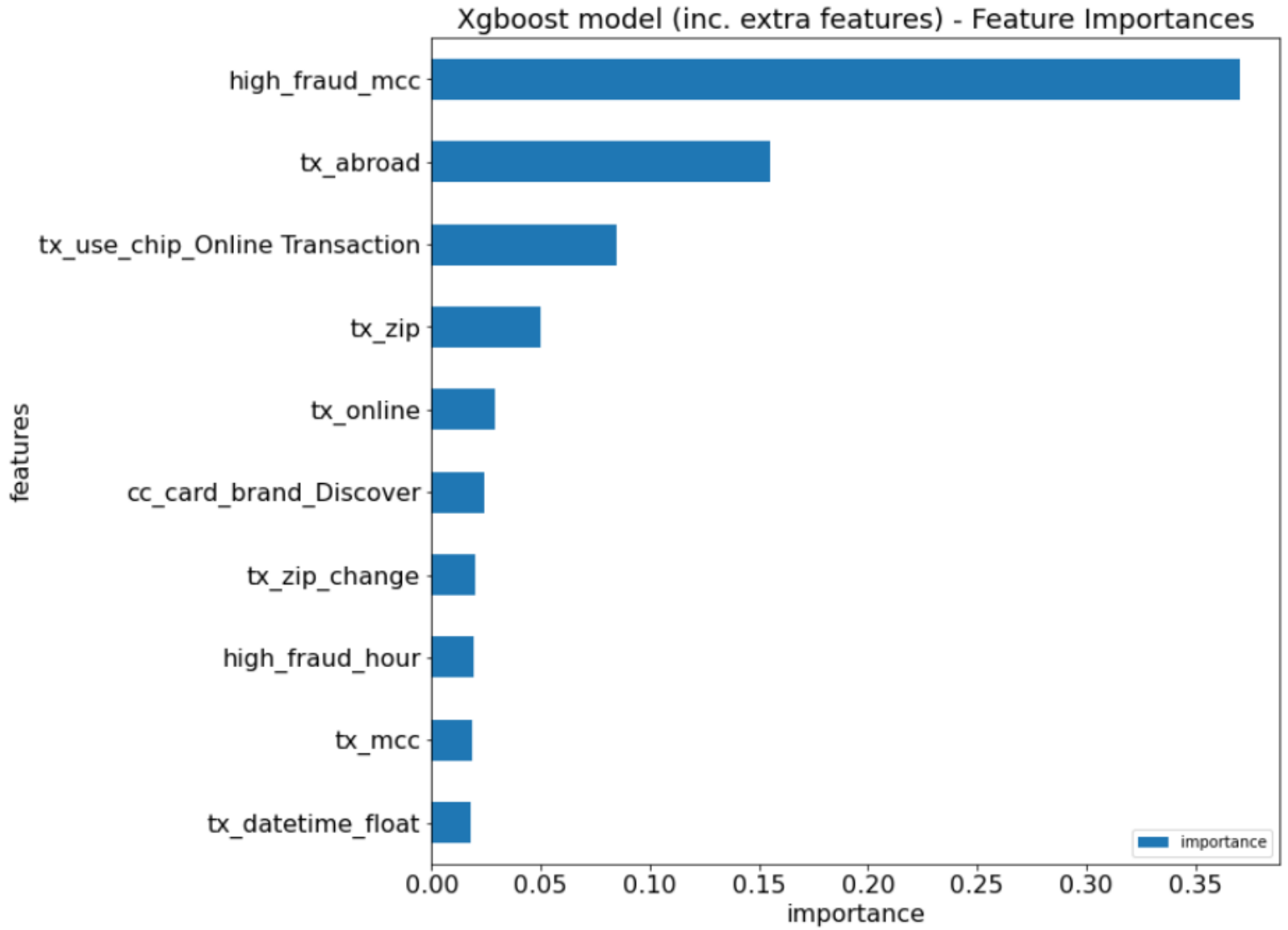
- With “standard” data set features
  - Mathematical best-fit model (linear regression)
  - Decision tree (optimised)
  - “Bagging” model - collections of decision trees each exposed to a subset of transactions and features
  - A “turbo charged” decision tree (XGBoost)
- Then the best of these with extra features
  - Some which add context (e.g. difference of transaction amount to running average)
  - Some built from EDA insights, e.g. does transaction have a merchant-charge-code associated with high levels of fraud?
- Then with the best model, how does changing the threshold at which we *call* a transaction a fraud transaction change our success factors?

Results

	Model	Model 'accuracy' (cf. baseline of 99.88%)	Percentage of fraud transactions detected	Percentage of all transactions incorrectly flagged as fraud
Standard features	Logistic regression (inc. CV)	87.26%	84.70%	12.73%
	Decision Tree (optimised parameters)	91.58%	91.60%	8.41%
	Decision Tree with bagging (80% features)	96.33%	94.80%	3.66%
	XGBoost	96.19%	97.40%	3.81%
Extra features	Decision Tree with bagging (80% features)	96.04%	95.20%	3.95%
	XGBoost	96.47%	98.50%	3.52%
	XGBoost - prob. threshold 0.84	98.35%	95.00%	1.64%
	XGBoost - prob. threshold 0.945	99.08%	90.10%	0.91%
	XGBoost - prob. threshold 0.975	99.41%	85.40%	0.57%
	XGBoost - prob. threshold 0.9986	99.88%	55.50%	0.07%

All results are against the test set (893k cases, 1k fraud)

Did our best model agree with out EDA?



Business implications/conclusions

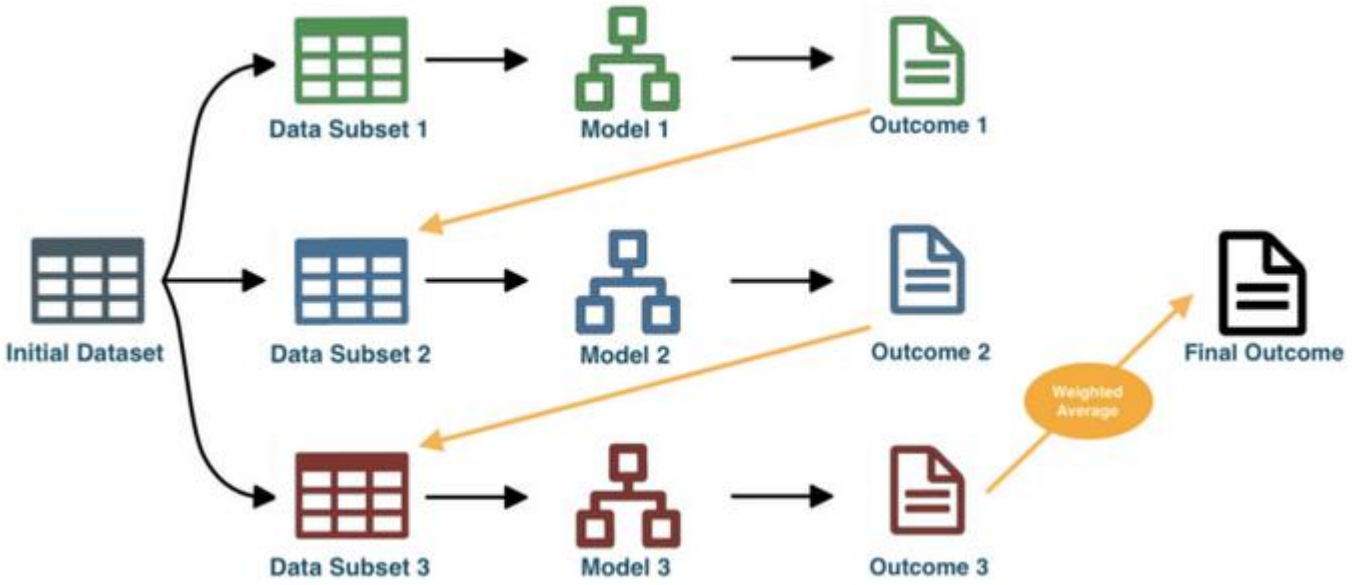
- Detecting credit-card fraud is a challenge for machine learning
- It is possible to detect nearly all fraud but this comes at a cost of many “false positives”, i.e. non-fraud transactions which we identify as fraud.
- Remedies?
  - Manual checks by back-office staff
  - Customer validation processes
  - Adjust the model threshold accepting the trade off, i.e.
    - Have less cases incorrectly flagged as fraud, but more fraud goes undetected.
- Ultimately it will be a cost-based decision.

Further work

- Detailed analysis of the misclassifications
- Better data
  - More? Cloud?
  - Oversampling?
  - Real data?
- Model improvements
  - (More) Extra features
  - Other model types
  - Parameter optimisation

Q&A

Appendix - XGBoost



- Models trained sequentially. Each model trained to correct the errors of the last. At each round, the errors in the last round(model) are given a heavier weight than the correct outcomes.
- Models are optimised using a gradient-descent algorithm with boosting so as to converge more quickly.
- A “loss” function is used to direct the optimisation, e.g. mean-squared error.
- Final classifier uses a weighted average of the ensemble of models created.