Credit Card Fraud Project

Michael Ramsey

Dataset Description

- The dataset contains transactions made by credit cards in September 2013 by European cardholders over the course of two days
- 492 transactions out of 284,807 have been identified as fraudulent
- There are 30 predictive features including ‘Amount’ (i.e., Euro amount of transaction), ‘Time’ (i.e., time since first transaction in dataset in seconds) and 28 others which are the result of a PCA transformation to maintain privacy
- The label is a binary feature where 1 corresponds to a fraud label

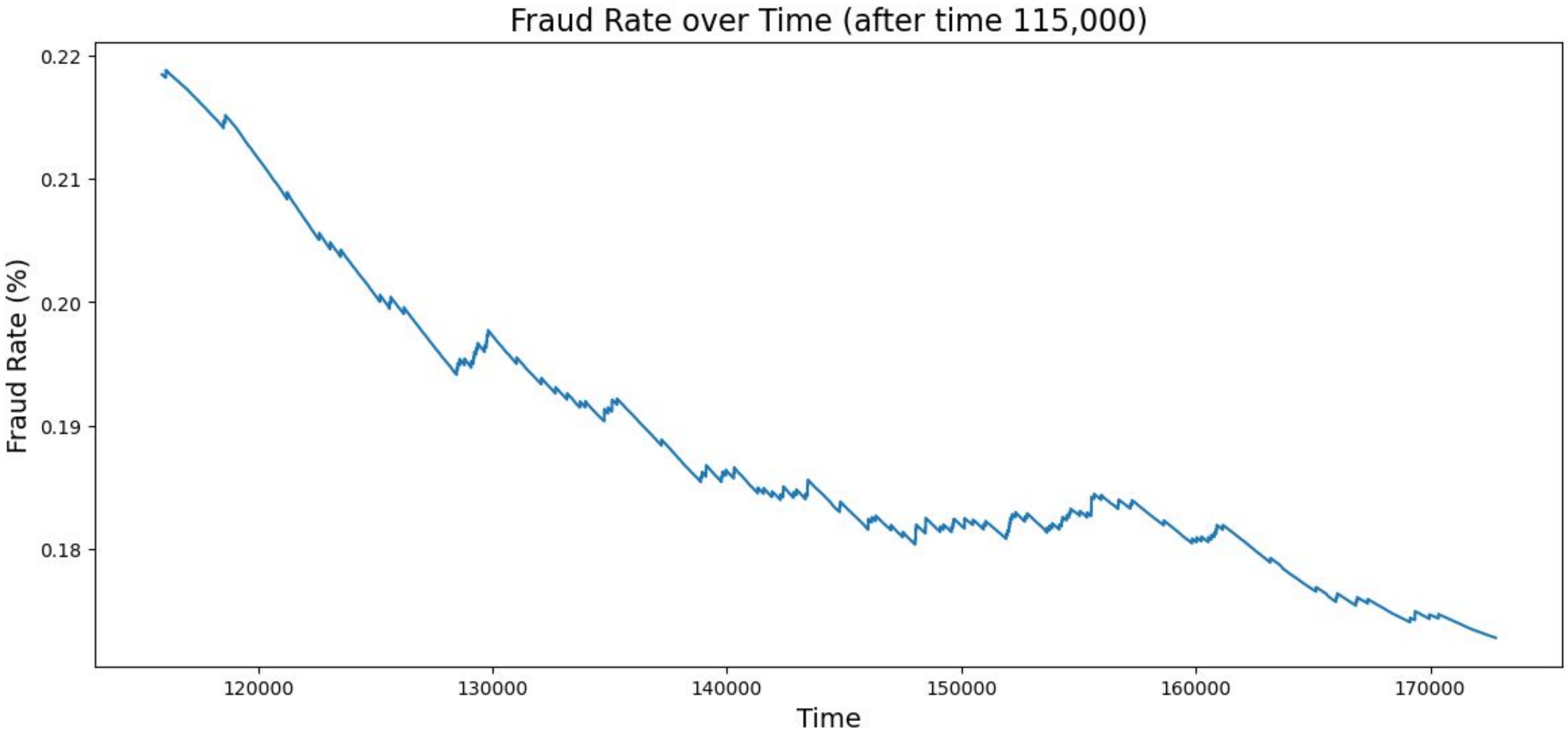
EDA: Key Takeaways (1)

- Fraud rate at time t (i.e., number of instances of fraud per transaction up to time t) generally decreased over the period in question after a spike at the beginning of the period; this suggests that the ‘Time’ feature should have some bearing on our fraud predictions as we expect less fraud at certain times than others (slide 5)
- Transaction amount distributions look very different for the fraudulent and non-fraudulent transaction populations; because approximately 40% of fraudulent transactions are under ~$1 as opposed to ~10% of non-fraudulent transactions, the ‘Amount’ feature can be used to separate the classes (slide 6); a Kolmogorov–Smirnov analysis validates this claim

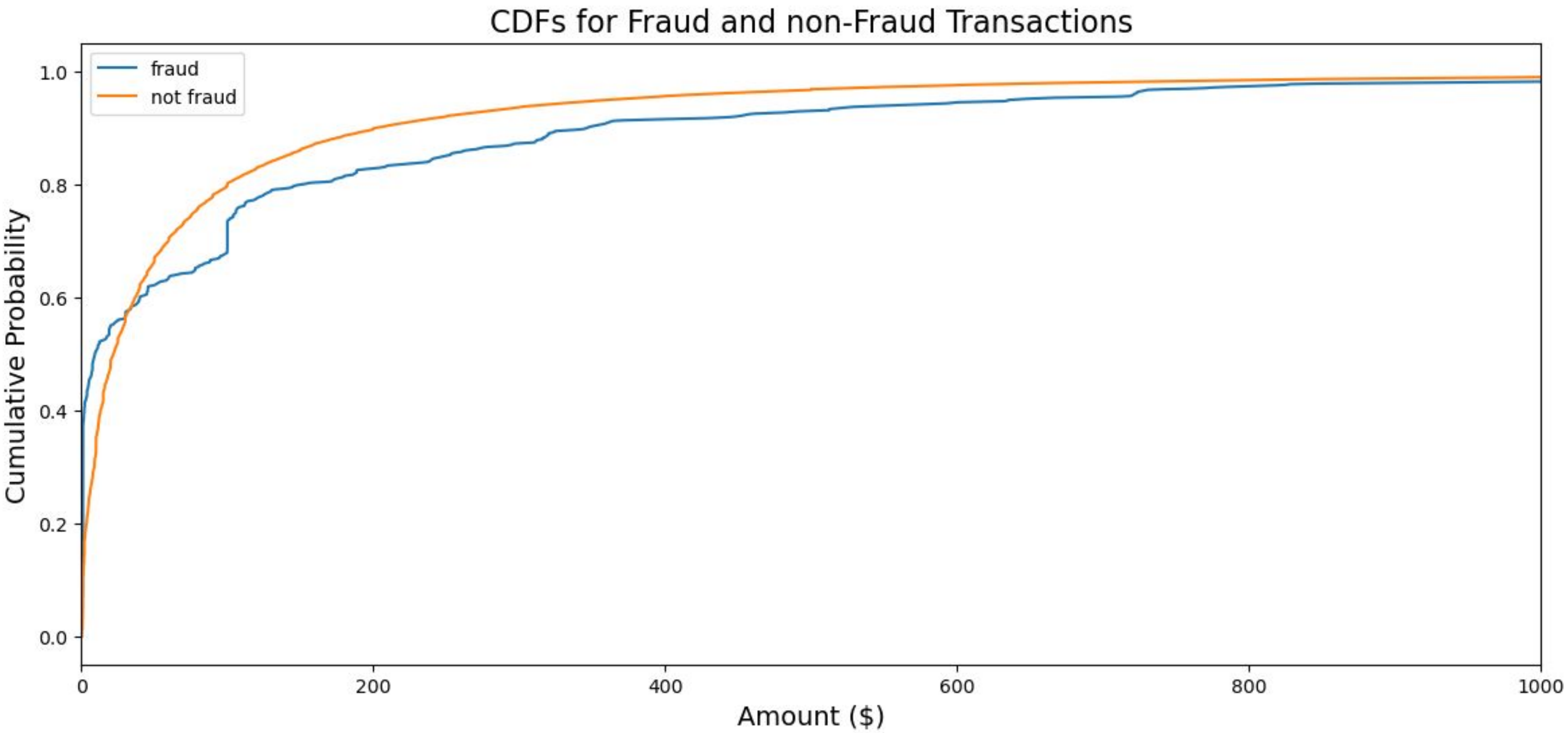
EDA: Key Takeaways (2)

- All PCA features seem to be able to at least partially separate the fraud and no-fraud classes based on differences in descriptive statistics for each class (slides 7, where rows correspond to features sorted by feature importance as determined by MRMR)

EDA: Key Takeaways



EDA: Key Takeaways



EDA: Key Takeaways

	fraud_mean	no_fraud_mean	fraud_median	no_fraud_median	fraud_min	no_fraud_min	fraud_max	no_fraud_max
0	-7.848295	0.013581	-6.243643	-0.076334	-29.626452	-20.131553	7.934890	10.895018
1	-7.272864	0.012586	-7.020407	0.054191	-20.044280	-19.186530	3.591116	10.981465
2	-6.264407	0.010840	-5.506937	0.141792	-18.698680	-15.157119	1.377043	7.854679
3	-5.213660	0.009022	-4.205202	-0.084376	-22.581908	-13.538252	3.702478	21.807579
4	-4.724610	0.008176	-4.051115	0.076892	-16.125344	-11.544131	3.583054	19.760439

Model

- Using all features with a training set of the first ~200k observations:
    1. Train a logistic regression model with class weights proportional to the class imbalance
    2. Train a multi-layer perceptron model with three hidden layers each comprised of 30 neurons using dropout=.2 and a loss function with class weights proportional to the class imbalance for 1000 epochs
    3. If 1) and 2) both predict fraud, then predict fraud
    4. If 1) or 2) predict fraud (but not both), then predict no fraud, but put in some queue for review
    5. Else predict no fraud

Results (on Test Set)

Model	Accuracy (%)	Precision (%)	Recall (%)
logistic regression	99.93	69.17	76.85
MLP	99.93	69.81	71.15
logistic regression + MLP	99.96	94.05	73.15

Takeaways

- Ensembling models of similar performance can lead to better than expected results due to the fact that models are learning different things
- In spite of the massive class imbalance, all models still get better accuracy than picking the majority class; this shows how doing something as simple as manipulating the loss function can pay huge dividends
- A simple model (logistic regression) can significantly outperform the standard model for anomaly detection (isolation forests) if the data is right
- More can be done to improve the ensemble/model (e.g., passing the data that the model is unsure about to another model that looks at different things, using data augmentation techniques such as adding random noise and/or oversampling the minority class, etc.)