Credit Card Fraud Detection

By: Xinkai Zhao, Yumeng Li

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Content

- Introduction
- Challenges
- Model Evaluation
- Conclusion

Background

Credit card industry grows bigger with the increasing popularity of electronic transactions.

Accurate fraud prevention system can help protect clients’ revenues.

Dataset description

- Vesta’s e-commerce transaction data
- 590,540 transactions, 394 features
    - Numerical: amount, Vesta features, …
    - Categorical: card type, purchaser email domain, issue bank, issue country, …

Content

- Introduction
- Challenges
  - Missing data
  - UID
  - Unbalanced Outcome Variable
- Model Evaluation
- Conclusion

Missing Data

- Delete 192 columns with >30% missing data
- Imputation:
    - Numerical: median
    - Categorical: mode

Content

- Introduction
- Challenges
  - Missing Data
  - UID
  - Unbalanced Outcome Variable
- Model Evaluation
- Conclusion

The Magic Feature - UID

- **Fraud status will impact following transactions with linked information**
- **Raw data does not have identifier for each card**

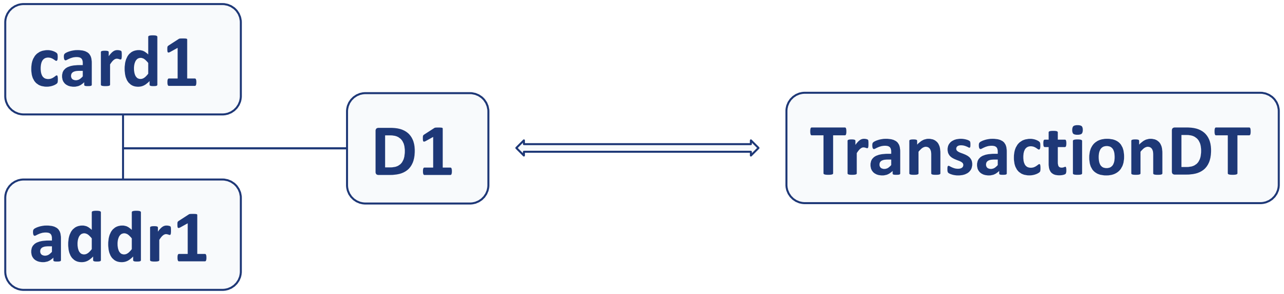
How do we classify fraudulent credit cards from only the transaction data?

Creating UID

Key features:

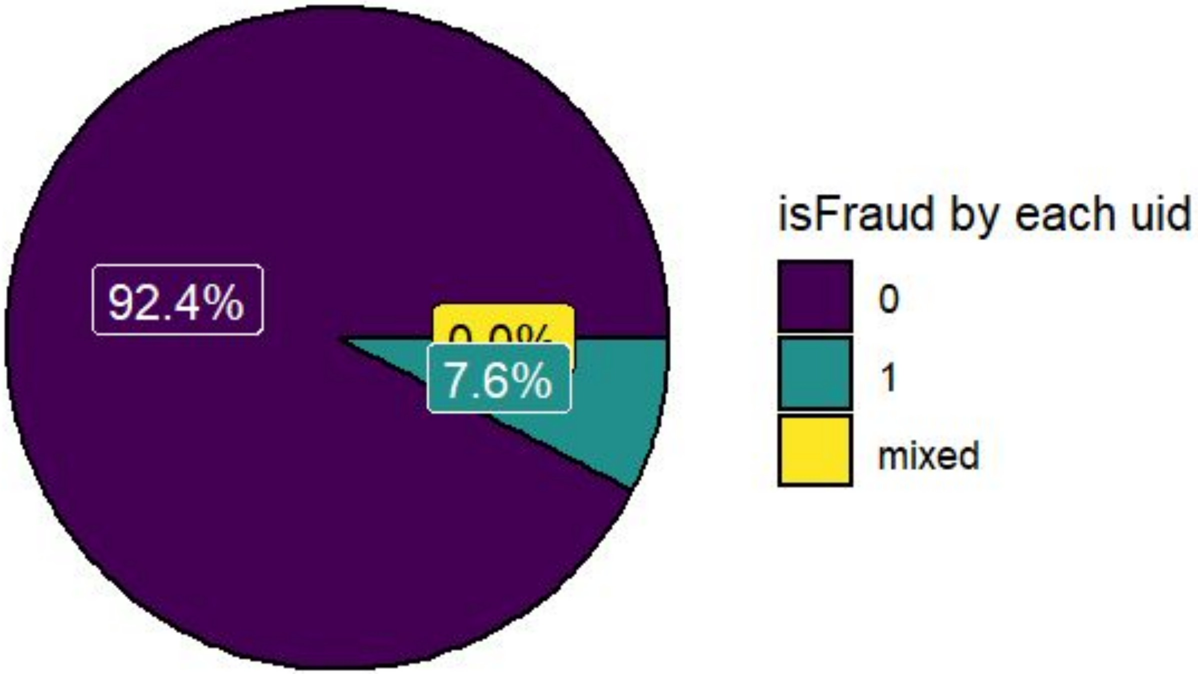
“card1”, “addr1”, “D1”, “TransactionDT”

Method:



https://www.kaggle.com/code/kyakovlev/ieee-uid-detection-v6

The Magic Feature - UID



The Magic Feature - UID

	TransactionID	isFraud	TransactionDT	TransactionAmt	card1	card4	addr1	D1	uid	DTdiff	D1diff
1261	2988261	1	129512	160.5	11839	visa	420.0	395.0	2988261.0	-0.0	0.0
1274	2988274	1	129834	280.0	11839	visa	420.0	395.0	2988261.0	-0.0	0.0
1282	2988282	1	130050	117.0	11839	visa	420.0	395.0	2988261.0	-0.0	0.0
127650	3114650	1	2537461	108.0	11839	visa	420.0	423.0	2988261.0	28.0	28.0
137995	3124995	1	2804429	171.0	11839	visa	420.0	426.0	2988261.0	31.0	31.0
230888	3217888	1	5474535	171.0	11839	visa	420.0	457.0	2988261.0	62.0	62.0
230893	3217893	1	5474733	100.0	11839	visa	420.0	457.0	2988261.0	62.0	62.0
316951	3303951	1	7889004	171.0	11839	visa	420.0	485.0	2988261.0	90.0	90.0
316955	3303955	1	7889277	117.0	11839	visa	420.0	485.0	2988261.0	90.0	90.0
341594	3328594	1	8429542	117.0	11839	visa	420.0	491.0	2988261.0	96.0	96.0
411332	3398346	1	10391596	117.0	11839	visa	420.0	514.0	2988261.0	119.0	119.0
411335	3398349	1	10391846	171.0	11839	visa	420.0	514.0	2988261.0	119.0	119.0
445894	3432916	1	11359563	117.0	11839	visa	420.0	525.0	2988261.0	130.0	130.0
479289	3466323	1	12438287	117.0	11839	visa	420.0	537.0	2988261.0	142.0	142.0
501966	3489000	1	13154560	171.0	11839	visa	420.0	546.0	2988261.0	151.0	151.0
501971	3489005	1	13154807	171.0	11839	visa	420.0	546.0	2988261.0	151.0	151.0

Day 2

Day 132

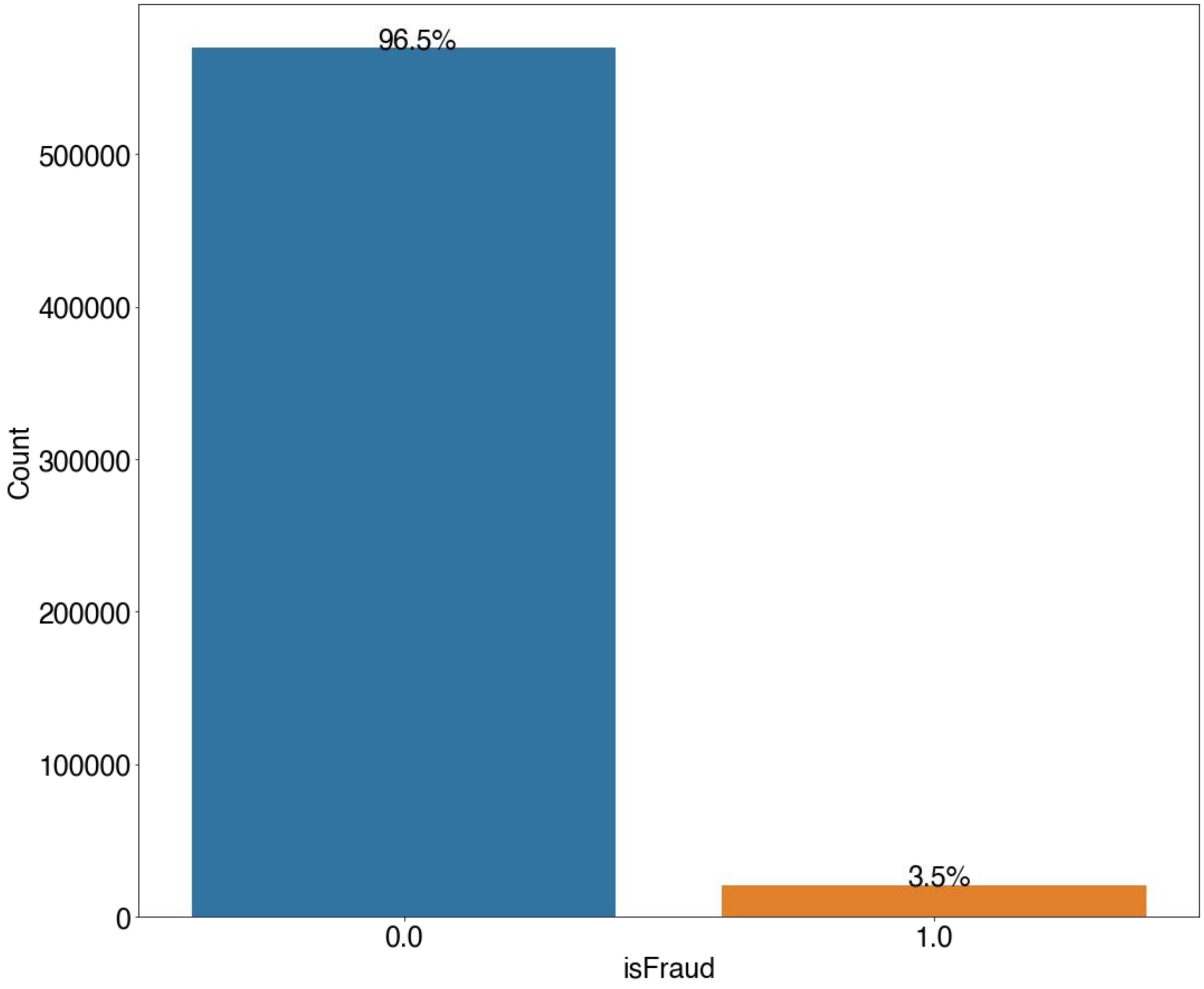
Preventing Overfitting

Aggregation:

- **Numerical: mean, standard deviation**
- **Categorical: n unique**

Content

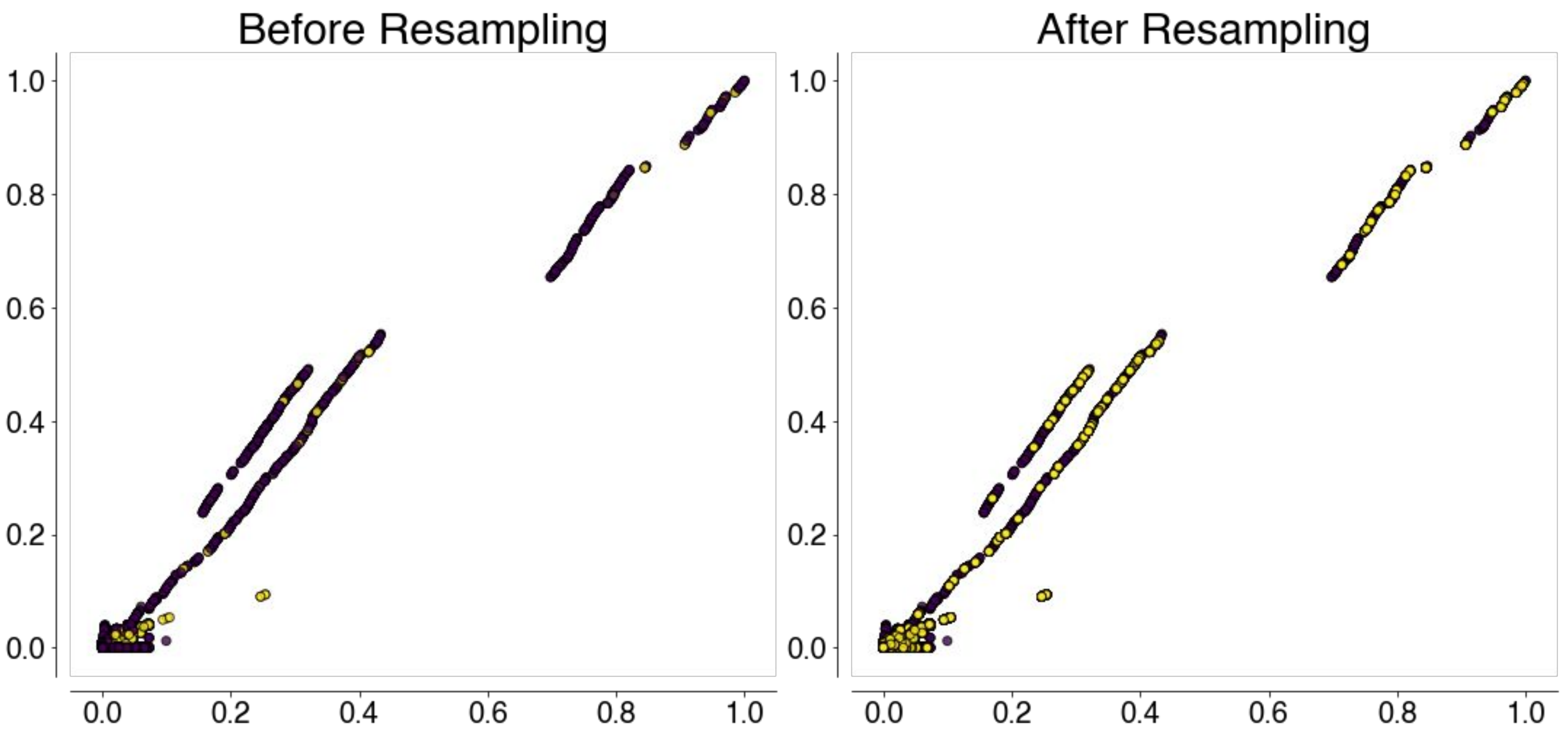
- Introduction
- Challenges
    - Missing Data
    - UID
    - Unbalanced Outcome Variable
- Model Evaluation
- Conclusion


96.5%
3.5%
500000
400000
300000
200000
100000
0
Count
0.0
1.0
isFraud

Random Over Sampler

- Object to over-sample the minority class
- Pick samples at random with replacement.

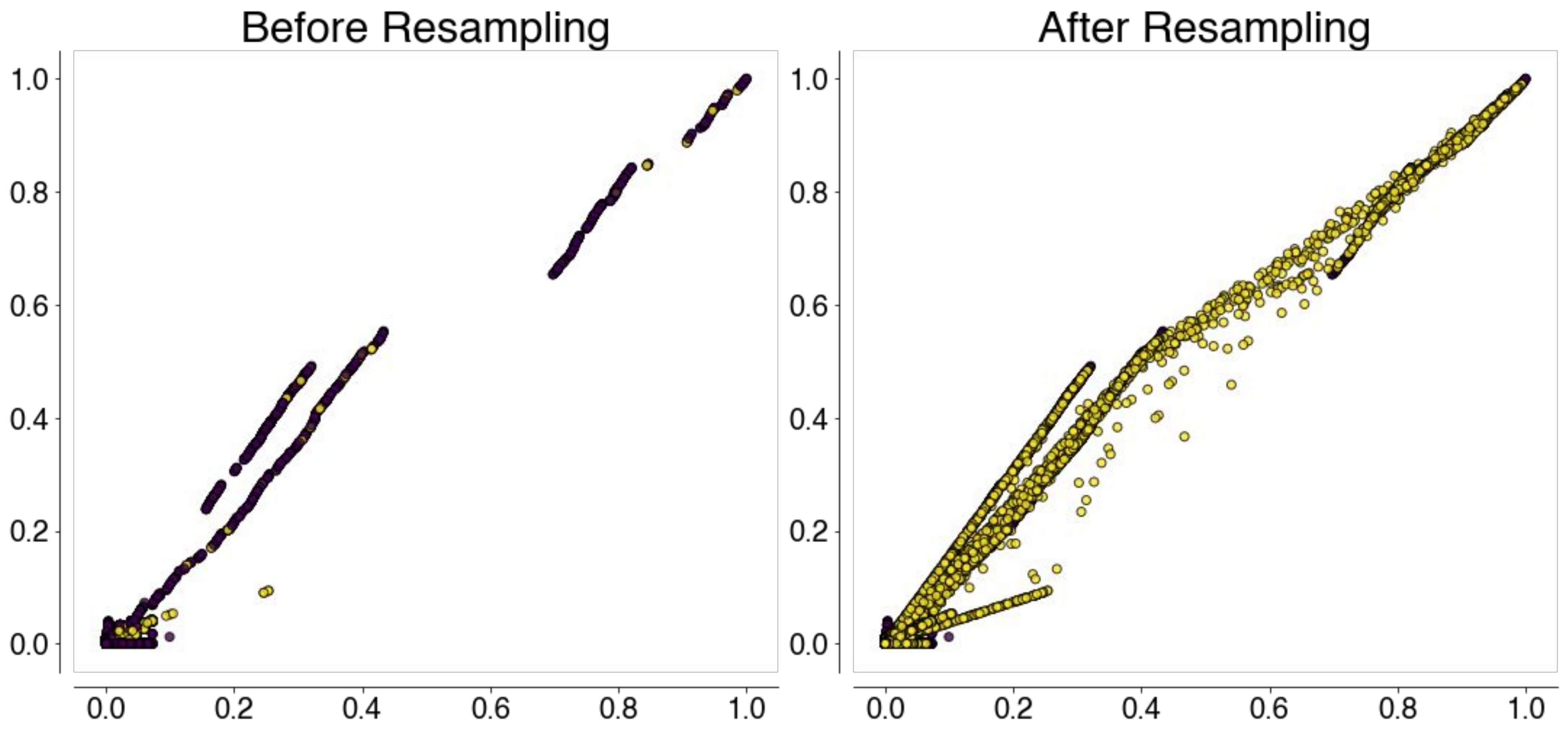
Random Over Sampler



SMOTE

- Over sampling technique
- Use a k-nearest neighbor algorithm to create synthetic data points
    - identify the minority class vector
    - compute a line between the minority data points and any of its neighbors and place a synthetic point
    - repeat until balanced

SMOTE



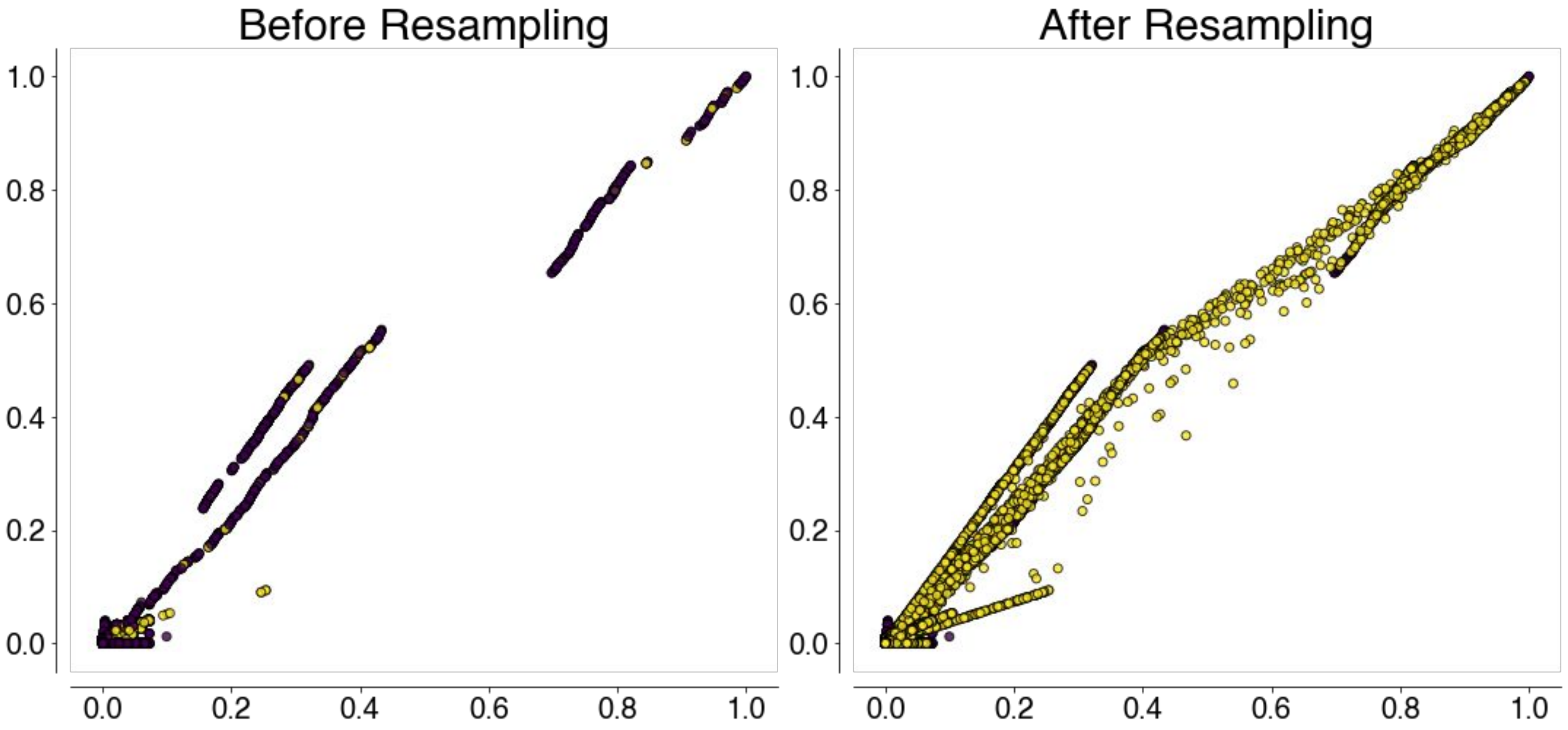
Before Resampling
After Resampling
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

SMOTE & Tomek Links

- SMOTE is applied to create new synthetic minority samples

- Tomek Links is used in removing the samples close to the boundary of the two classes, increase the separation

SMOTE & Tomek Links



Content

- Introduction
- Challenges
- Model Evaluation
- Conclusion

Evaluation Metrics

- Accuracy

  percentage of correctly classified

- Recall

  proportion of actual fraud is detected

- AUC

  how well the separation is

Evaluate KNN Algorithm

Resampling Methods	Accuracy	Recall	AUC
None	0.9675	0.1037	0.5513
Random Over Sampler	0.844	0.54	0.698
SMOTE	0.7162	0.72	0.7178
SMOTE + Tomek	0.7285	0.71	0.7183

Evaluate Random Forest

Resampling Methods	Accuracy	Recall	AUC
None	0.9742	0.29	0.644
Random Over Sampler	0.9825	0.73	0.861
SMOTE	0.9815	0.62	0.8088
SMOTE + Tomek	0.9815	0.62	0.8076

Content

- Introduction
- Challenges
- Model Evaluation
- Conclusion

Conclusion

- **PCA will decrease the performance of Random Forest**

Model	Accuracy	Recall	AUC
With PCA	0.7905	0.77	0.78
Without PCA	0.9825	0.73	0.861

Conclusion

- Choice of resampling depends on models
- Drawback of resampling methods: overfitting, hurt accuracy
- KNN model is a lazy algorithm

Thank You