RESEARCH OF BOOSTING ALGORITHMS VERSUS TRADITIONAL METHODS IN CREDIT CARD FRAUD DETECTION ACROSS VARIED DATASETS

Justs Viduss

Transport and Telecommunication Institute

Why This Research Matters

- **Addressing a Major Challenge**: This research directly addresses the critical need for more effective fraud detection systems in the banking sector, particularly in handling the complexity and volume of modern financial transactions.
- **Addressing Real-World Data Challenges**: Financial fraud detection faces the challenge of dealing with highly unbalanced datasets where fraudulent transactions are much less frequent than legitimate ones. The research explores the effectiveness of these algorithms in such settings, offering potential solutions to one of the biggest obstacles in fraud detection
- **Improvement Over Existing Methods**: By conducting a thorough comparison of boosting algorithms against traditional methods, this research fills a gap in the existing literature that often lacks detailed comparative analysis.

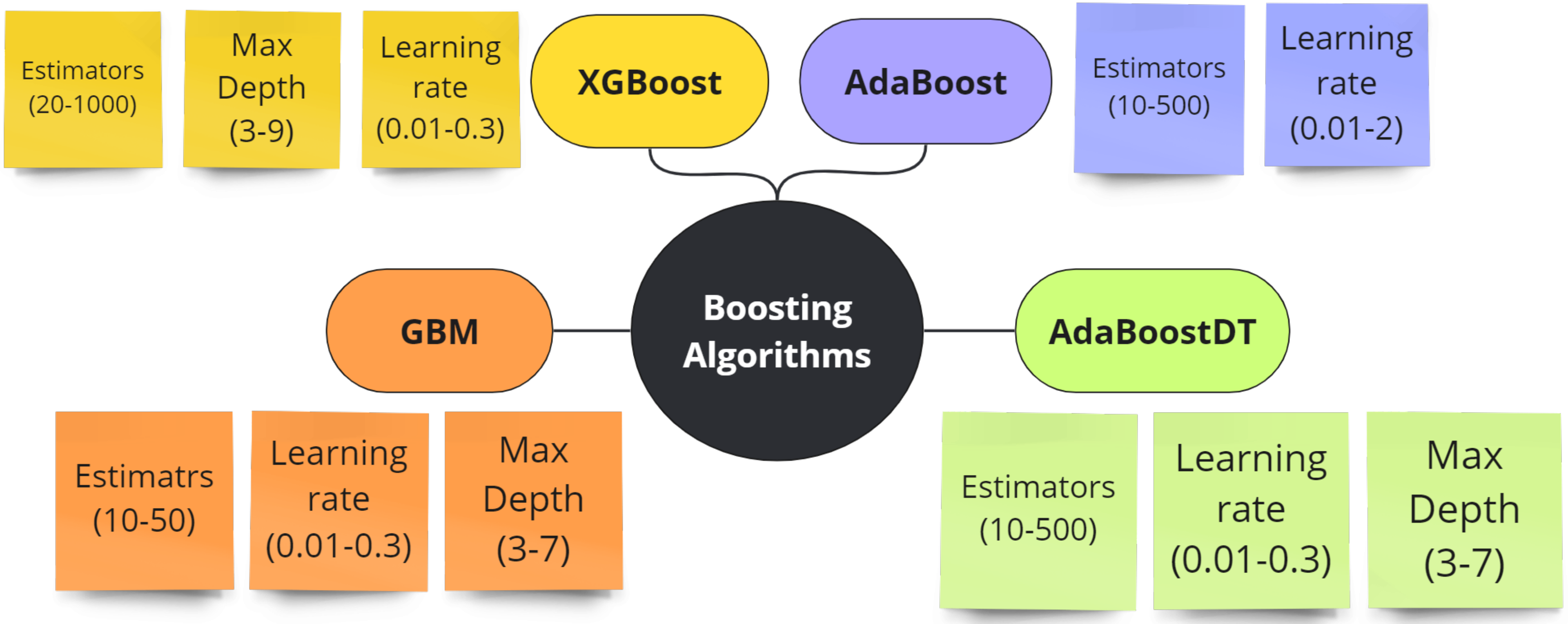
Objectives

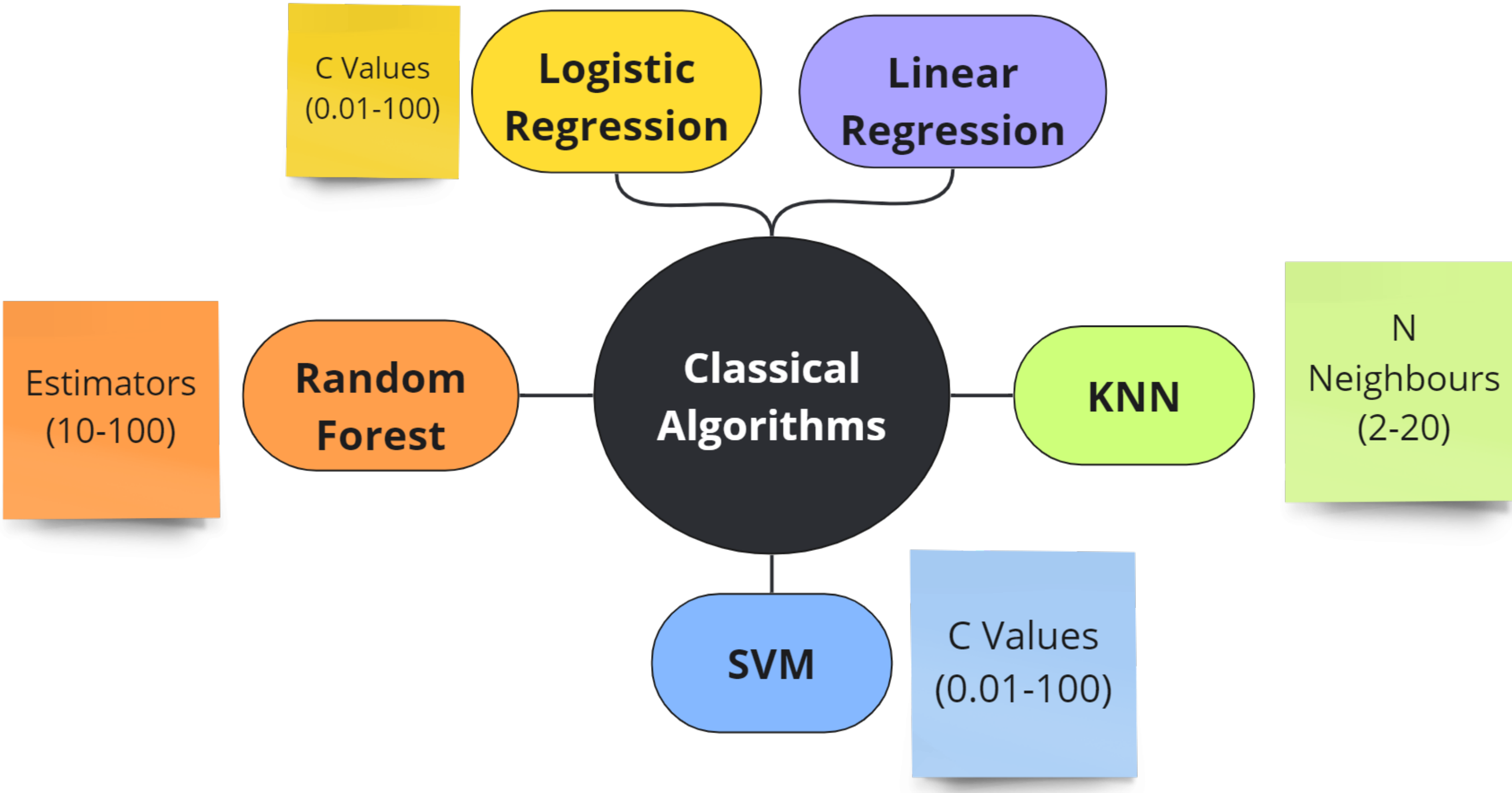
- To explore the evolution of fraud detection tactics in the banking industry
- To compare the efficacy of traditional and boosting algorithms in identifying credit card fraud using criteria such as accuracy, precision, recall, and the F1 score.
- To evaluate the computational efficiency of these algorithms, take into account parameters such as training time and resource use, which are crucial for real-time fraud detection.

Outline

- Algorithms tested
- Datasets
- Implementation
- Algorithm performance analysis
- Conclusions

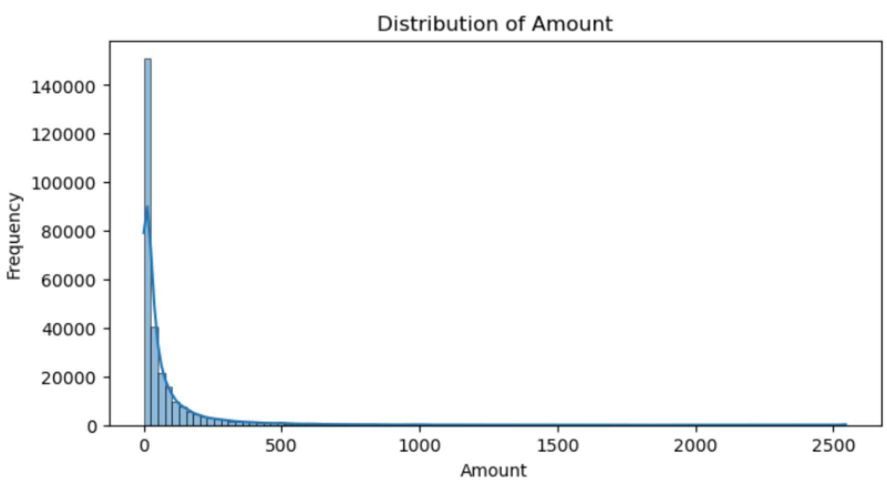
Algorithms tested

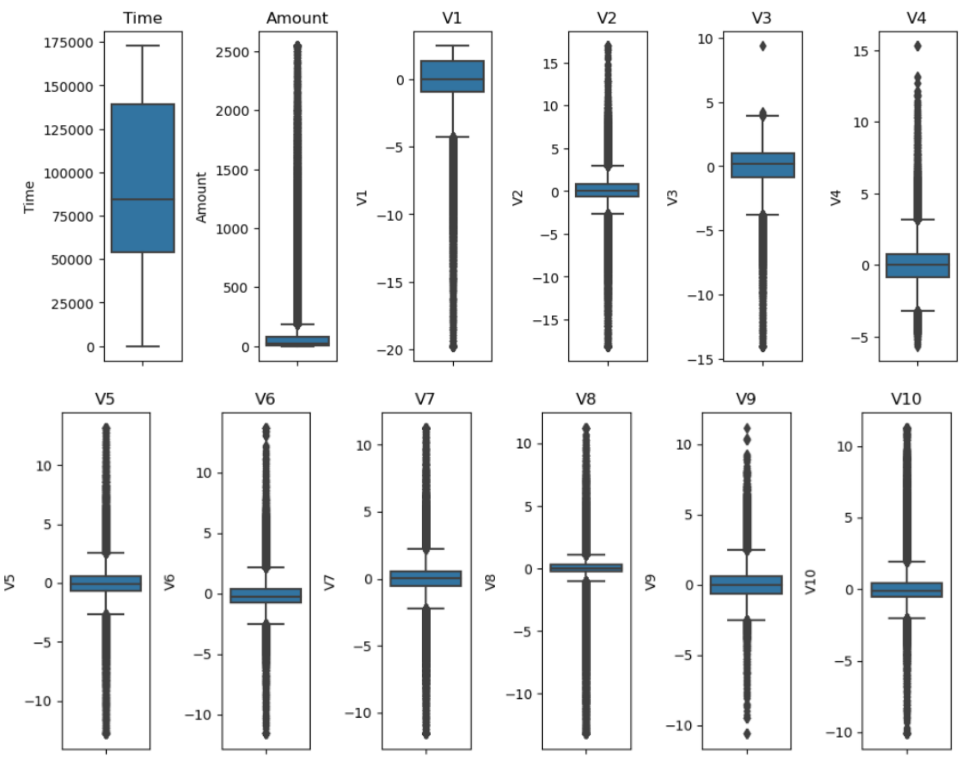



C Values
(0.01-100)
Logistic Regression
Linear Regression
Estimators
(10-100)
Random Forest
Classical Algorithms
KNN
N
Neighbours
(2-20)
SVM
C Values
(0.01-100)

Datasets

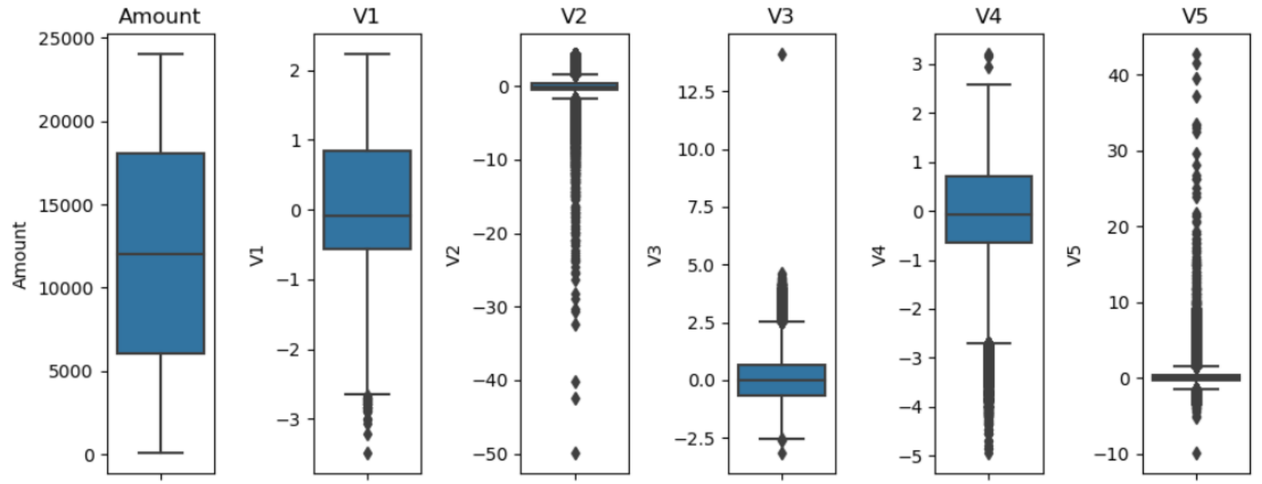
- **Unballanced** dataset: 284k rows, 30 features, 492 fraud cases (0.17%). All features exept time and transaction amount are anonymized using PCA. Source: Kaggle (2018) *transactions made by European cardholders in September 2013*

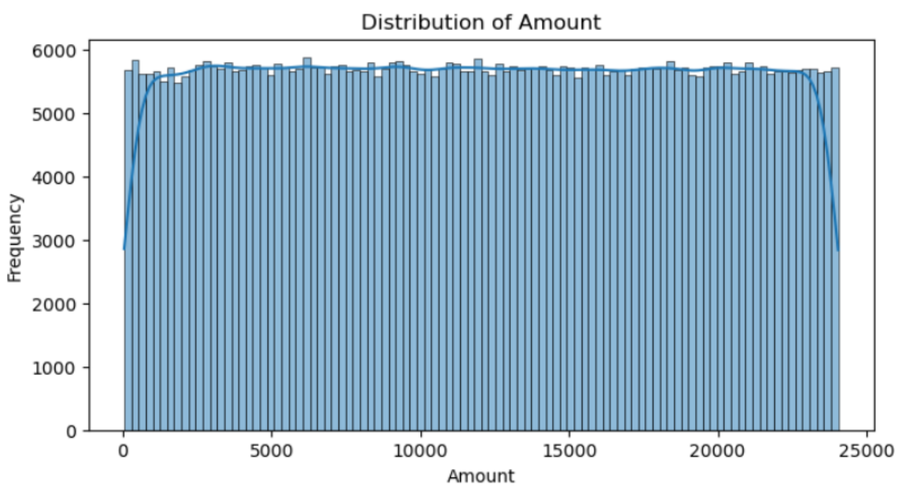


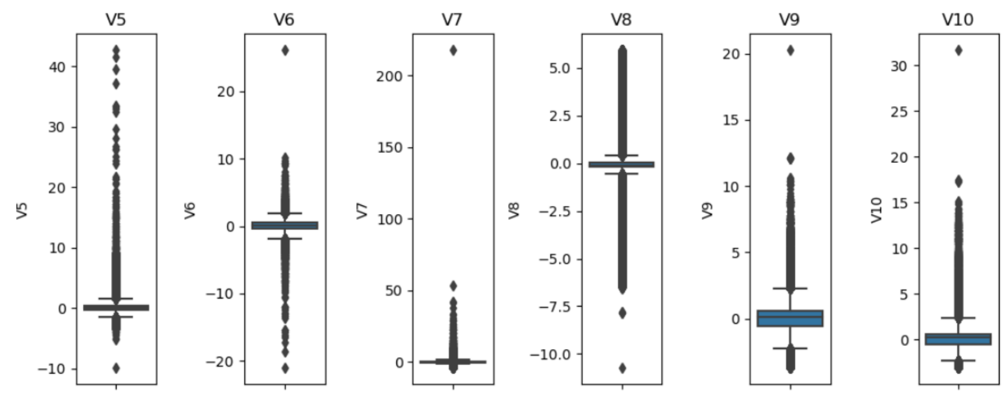


- **Ballanced** dataset: 568k rows, 29 features, 280k fraud cases (50%).

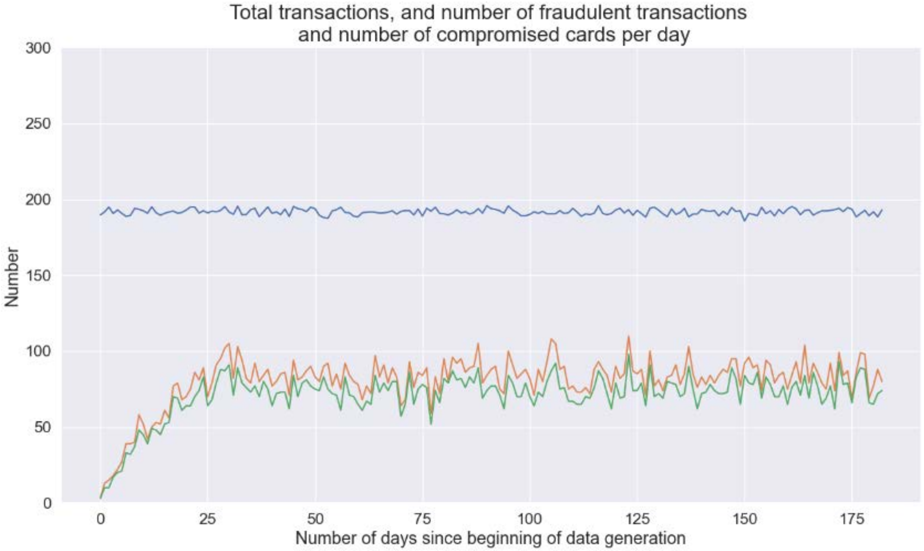
Source: Kaggle (2023). Transactions made by European cardholders in 2023

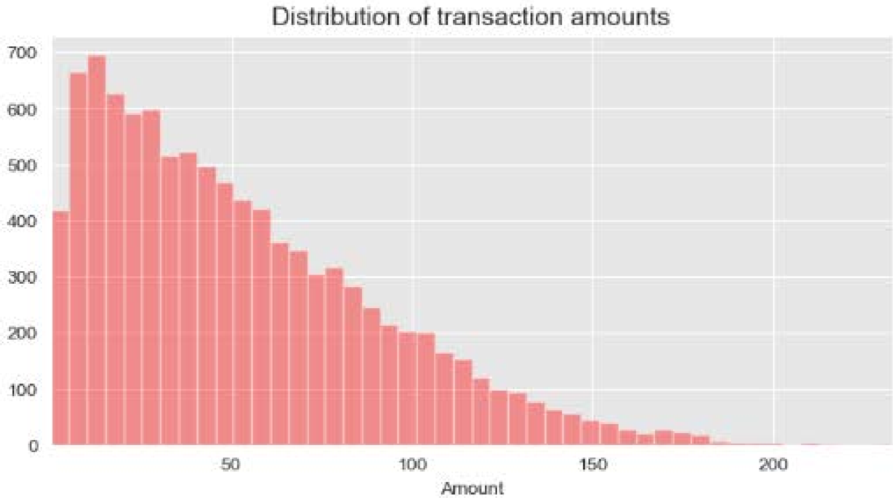


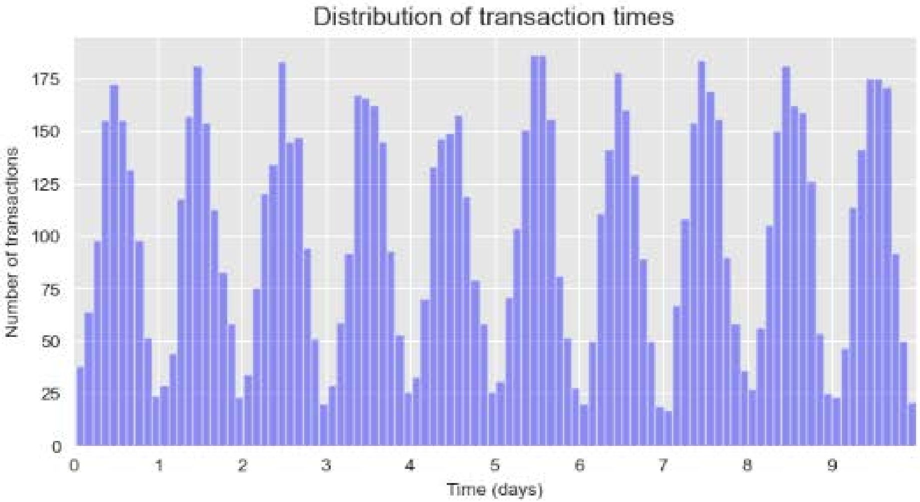




- **Synthetic** data: Generated with python script, 1.7m rows, 5 features, 14k fraud cases (0.84%)







Implementation

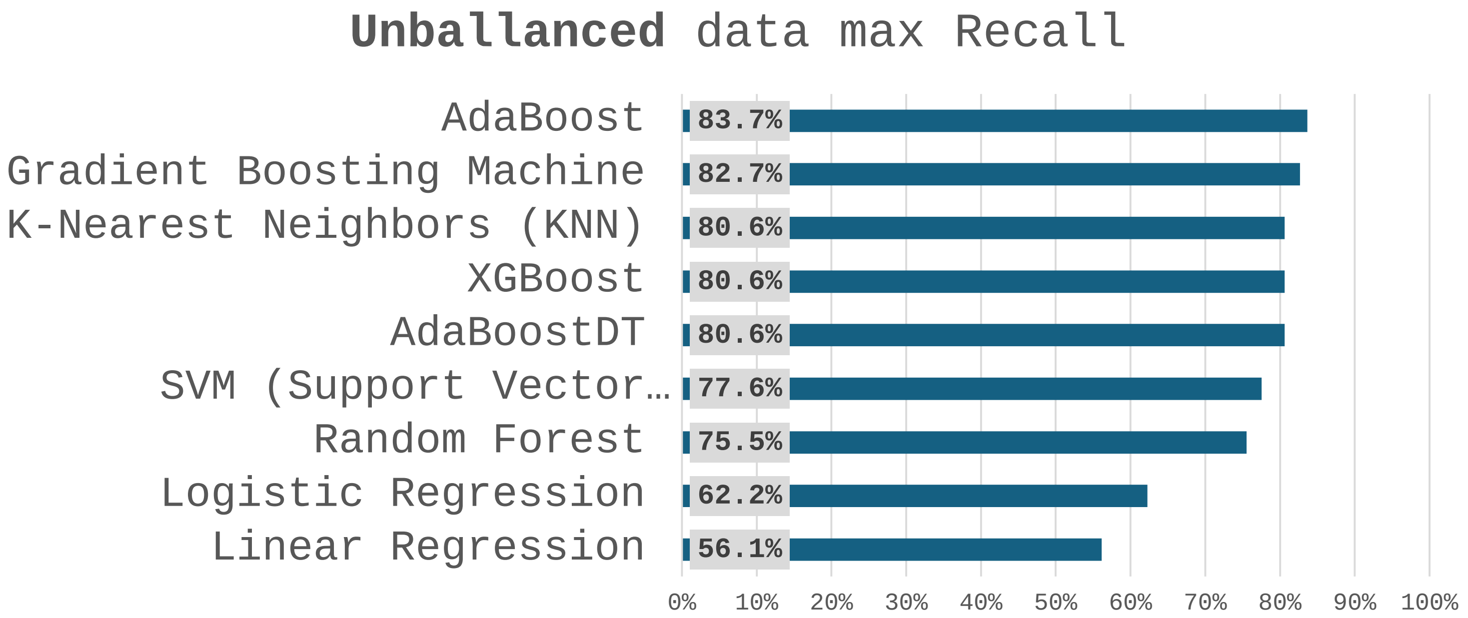
- Tools: Python notebooks, Excel
- Preprocessing: Data cleaning and preparation for modeling
- Parameter tuning: In total 900 semi-automated tests were done
- Testing environment: Laptop
  - 10 Core 12th gen Intel CPU
  - 16 GB RAM

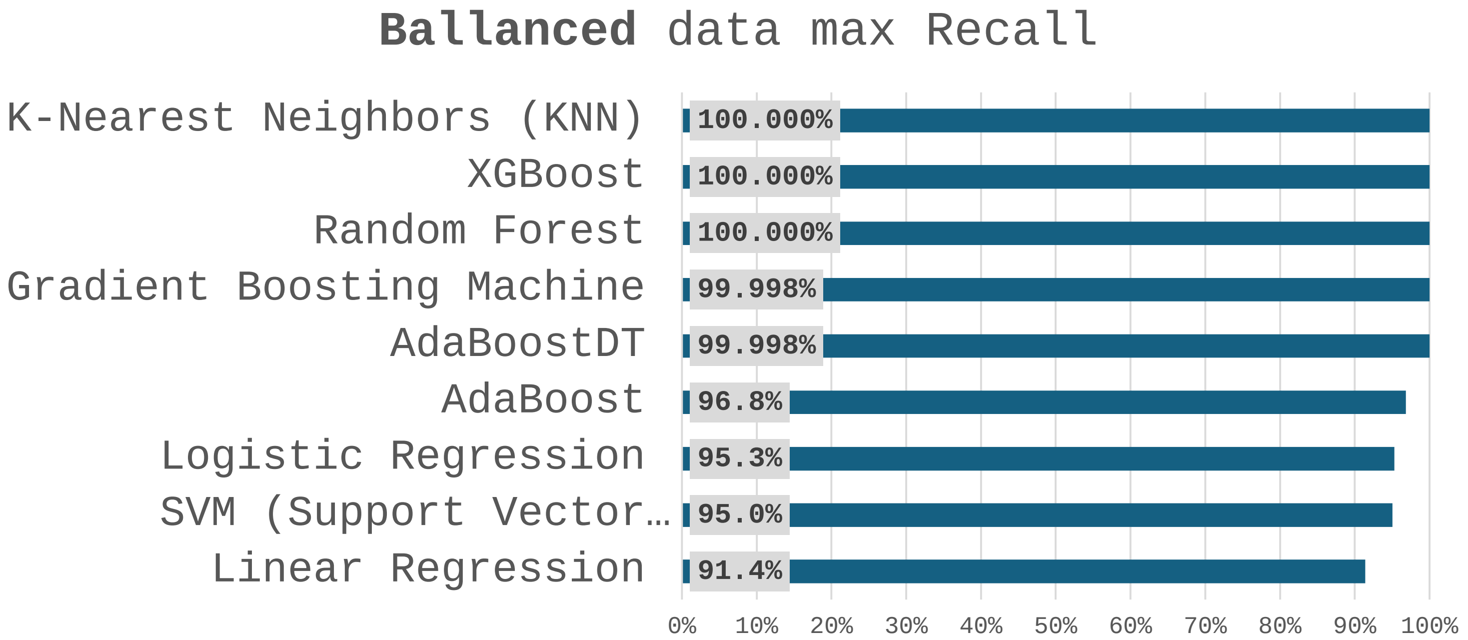
Algorithm performance analysis

Why each metric was chosen for tests:

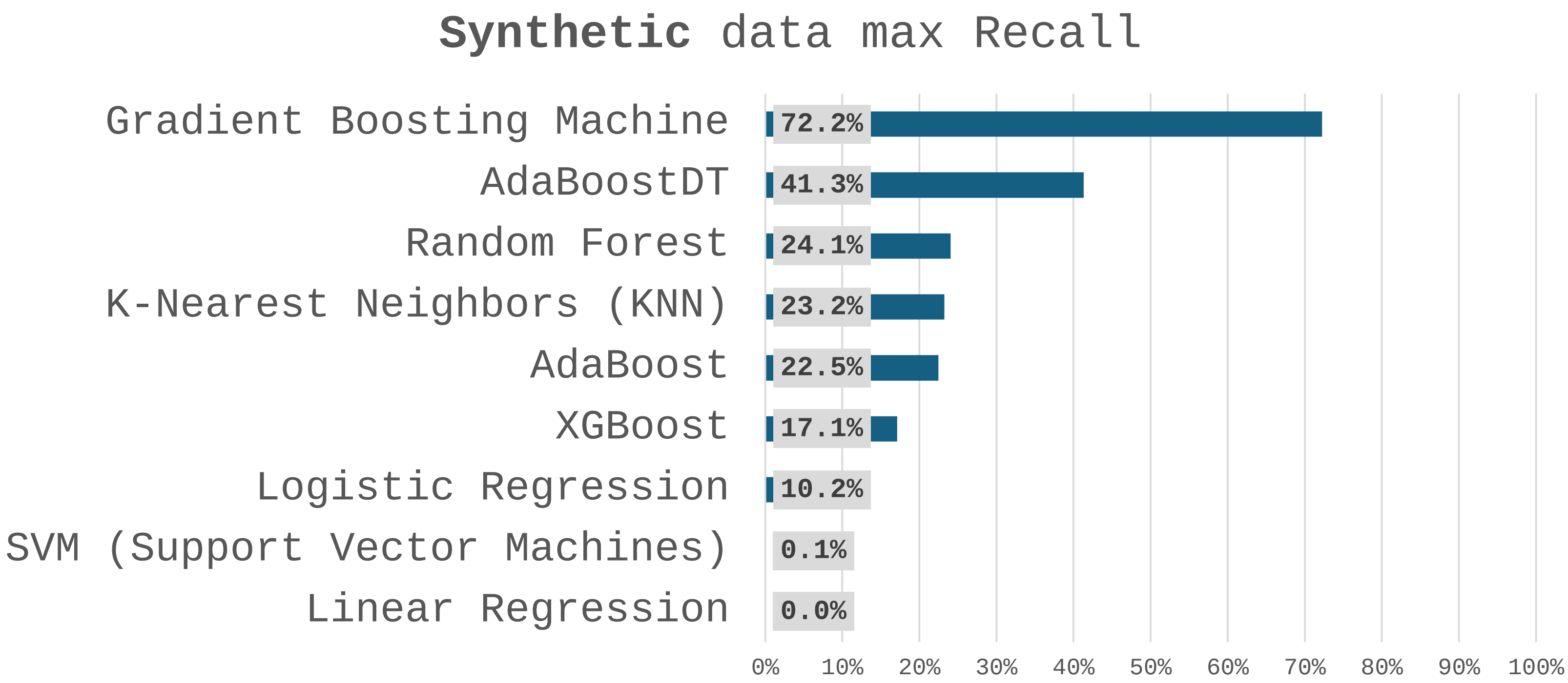
- **Recall** when it is critical to catch as many frauds as possible (minimizing false negatives)
- **Precision** when it is crucial to be as accurate as possible in your fraud predictions (minimizing false positives)
- **F1 Score** when you need a balance between precision and recall, and both types of errors are similarly costly
- **Accuracy** only when the classes are somewhat balanced or when you want a general idea of the model's performance across all predictions

Recall


Unballanced data max Recall
AdaBoost 83.7%
Gradient Boosting Machine 82.7%
K-Nearest Neighbors (KNN) 80.6%
XGBoost 80.6%
AdaBoostDT 80.6%
SVM (Support Vector… 77.6%
Random Forest 75.5%
Logistic Regression 62.2%
Linear Regression 56.1%
0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%


Ballanced data max Recall
K-Nearest Neighbors (KNN) 100.000%
XGBoost 100.000%
Random Forest 100.000%
Gradient Boosting Machine 99.998%
AdaBoostDT 99.998%
AdaBoost 96.8%
Logistic Regression 95.3%
SVM (Support Vector… 95.0%
Linear Regression 91.4%
0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Recall


Synthetic data max Recall
Gradient Boosting Machine
AdaBoostDT
Random Forest
K-Nearest Neighbors (KNN)
AdaBoost
XGBoost
Logistic Regression
SVM (Support Vector Machines)
Linear Regression
72.2%
41.3%
24.1%
23.2%
22.5%
17.1%
10.2%
0.1%
0.0%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%

Accuracy

	Max of Accuracy		
	Unballanced	**Ballanced**	**Synthetic**
XGBoost	99,96%	99,99%	99,26%
AdaBoostDT	99,96%	99,98%	99,50%
Random Forest	99,96%	99,99%	99,38%
K-Nearest Neighbors (KNN)	99,95%	99,93%	99,36%
AdaBoost	99,95%	97,62%	99,37%
Gradient Boosting Machine	99,95%	99,96%	99,60%
SVM (Support Vector Machines)	99,94%	96,39%	99,19%
Logistic Regression	99,92%	96,53%	99,27%
Linear Regression	99,91%	94,84%	99,19%

F1 Score

	Max of F1 Score		
	Unballanced	**Ballanced**	**Synthetic**
XGBoost	88,27%	99,99%	26,86%
AdaBoostDT	87,64%	99,98%	57,55%
Random Forest	85,55%	99,99%	38,74%
K-Nearest Neighbors (KNN)	85,39%	99,93%	36,02%
AdaBoost	85,08%	97,61%	36,72%
Gradient Boosting Machine	82,76%	99,96%	72,04%
SVM (Support Vector Machines)	82,16%	96,34%	0,21%
Logistic Regression	72,62%	96,49%	18,47%
Linear Regression	67,48%	94,67%	0,07%

Precision

	Max of Precision		
	Unballanced	**Ballanced**	**Synthetic**
Random Forest	100,0%	100,0%	99,6%
Gradient Boosting Machine	100,0%	99,9%	100,0%
XGBoost	98,7%	100,0%	100,0%
AdaBoostDT	98,7%	100,0%	100,0%
K-Nearest Neighbors (KNN)	95,0%	99,9%	99,2%
AdaBoost	92,8%	98,5%	100,0%
SVM (Support Vector Machines)	89,2%	98,1%	100,0%
Logistic Regression	87,1%	98,2%	100,0%
Linear Regression	84,6%	98,2%	100,0%

Conclusions

- For Fraud detection in real world scenarious AdaBoost has best performance from tested algorithms, and outperforms other Boosting and classical algorithms
- Other algorithms perform better in different situations which is interesting for other industries where Accuracy, F1 Score or Precision would be more important
- Gradient Boosting Machine might be go to algorithm with uncertain or changing data, as it performed well on both, real and synthetic data

Acknowledgements

- The research was supervised by Dr.sc.ing., Professor Nadezda Spiridovska, whose insights and guidance were invaluable throughout the study

Thank you

Justs Viduss

viduss.justs@gmail.com