Team 9 BA810

12/7/2022

Fraud Detection

Chuheng Yu, Linting Wang
Yuhan Wang, Tyler Christoforo

Notebook link:
https://colab.research.google.com/drive/1fk3YjF9j8xl893SzFtUWxx75jpKv8y6C?usp=sharing

Recap

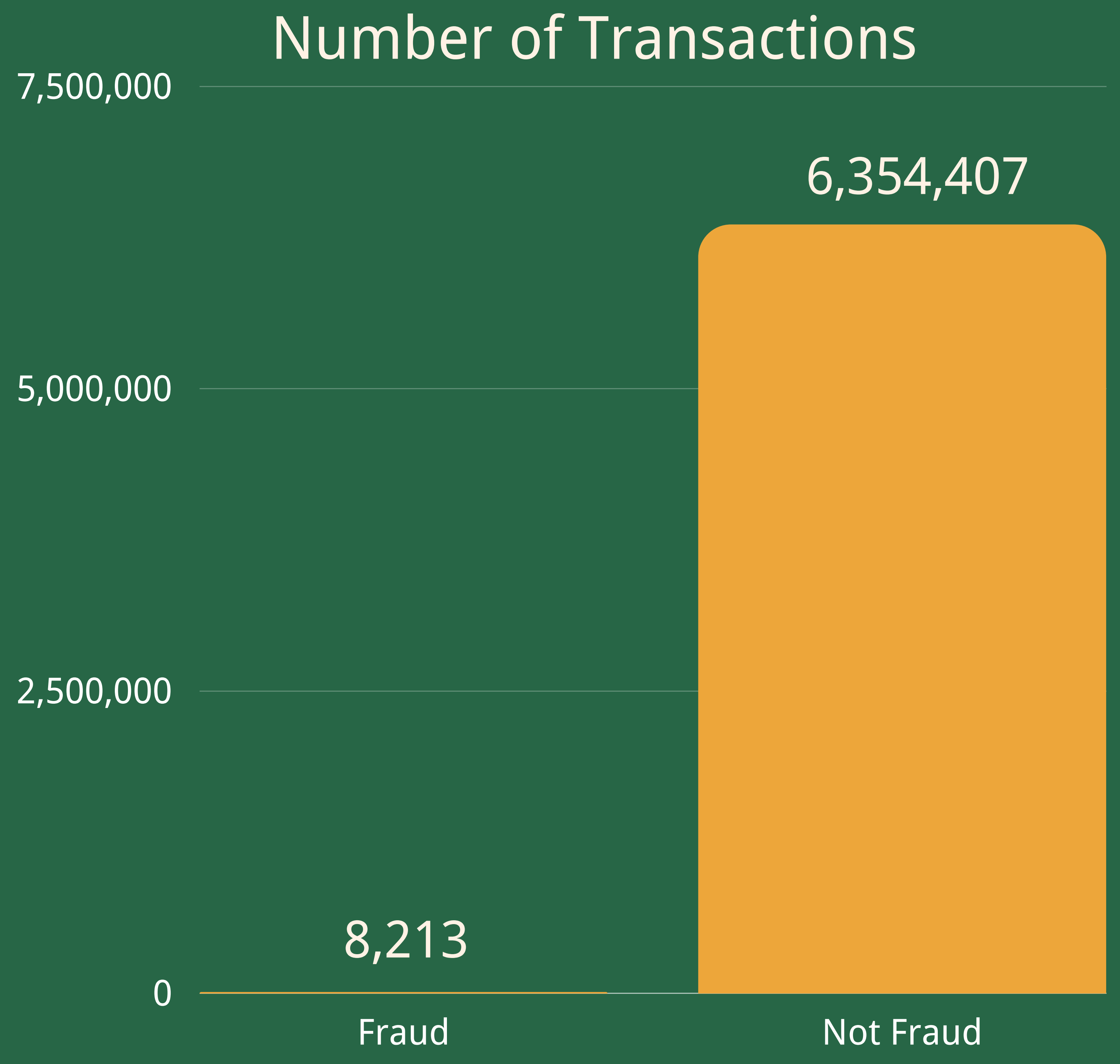
- Objective:
    - Determine online fraudulent payment patterns within existing dataset
- Data Source:
    - From Medium.com made public by Edgar Alonso Lopez-Rojas
- Columns within the dataset:
    - String: nameOrig, nameDest
    - Numerical: step, amount, oldbalanceOrg, newbalanceOrig, oldbalanceDest, newbalanceDest
    - Categorical: type
    - Target column: isFraud

Problems to resolve

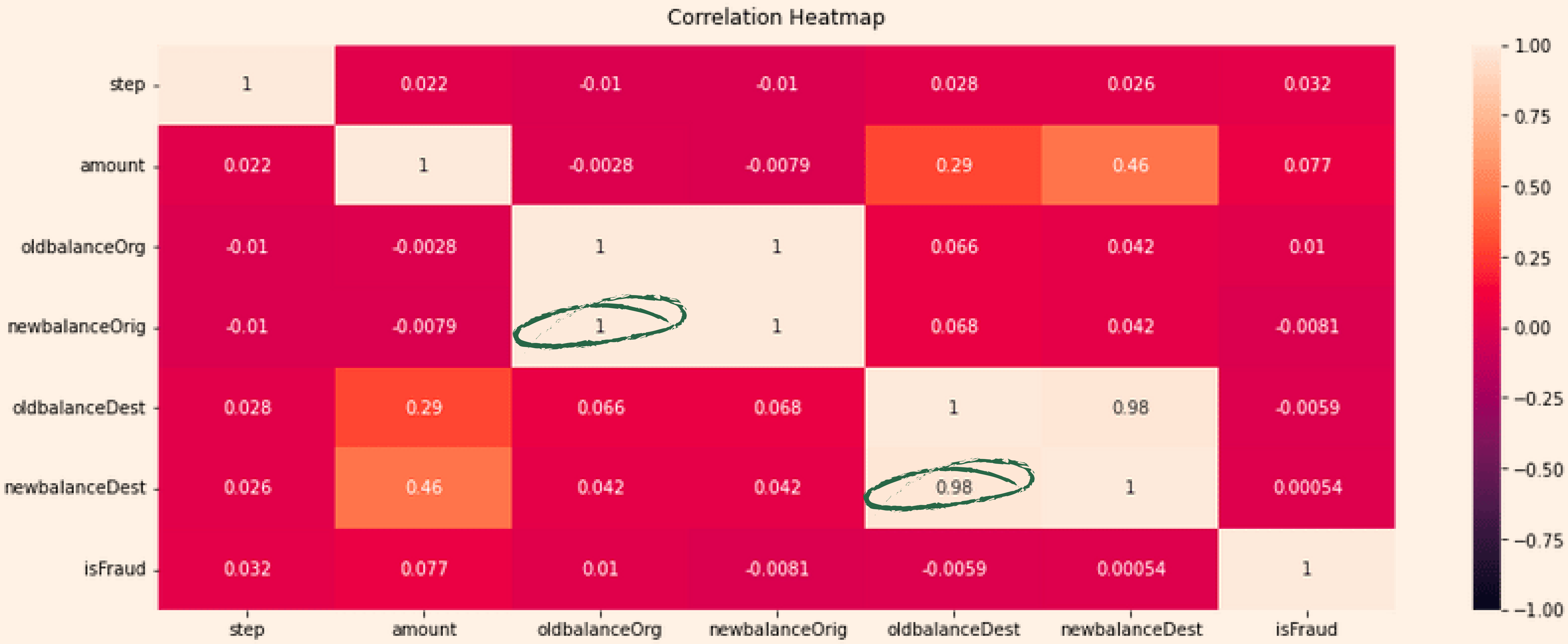
Data imbalance

Danger of overfitting

Correlation between columns



Correlation Heatmap



	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud
step	1	0.022	-0.01	-0.01	0.028	0.026	0.032
amount	0.022	1	-0.0028	-0.0079	0.29	0.46	0.077
oldbalanceOrg	-0.01	-0.0028	1	1	0.066	0.042	0.01
newbalanceOrig	-0.01	-0.0079	1	1	0.068	0.042	-0.0081
oldbalanceDest	0.028	0.29	0.066	0.068	1	0.98	-0.0059
newbalanceDest	0.026	0.46	0.042	0.042	0.98	1	0.00054
isFraud	0.032	0.077	0.01	-0.0081	-0.0059	0.00054	1

Data preparation

Column examination

Check Null values

Check correlation

Check patterns of fraud

Check column types

Feature engineering

errorDest = amount + oldbalancedest - newbalancedest

errorOrig = amount + newbalanceorig - oldbalanceorig

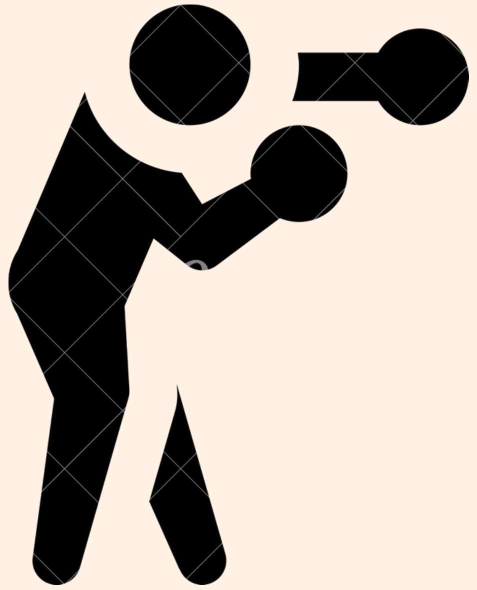
transferPercentage = amount/oldbalanceorig x 100

Detail management

Drop rows with 0 as the amount

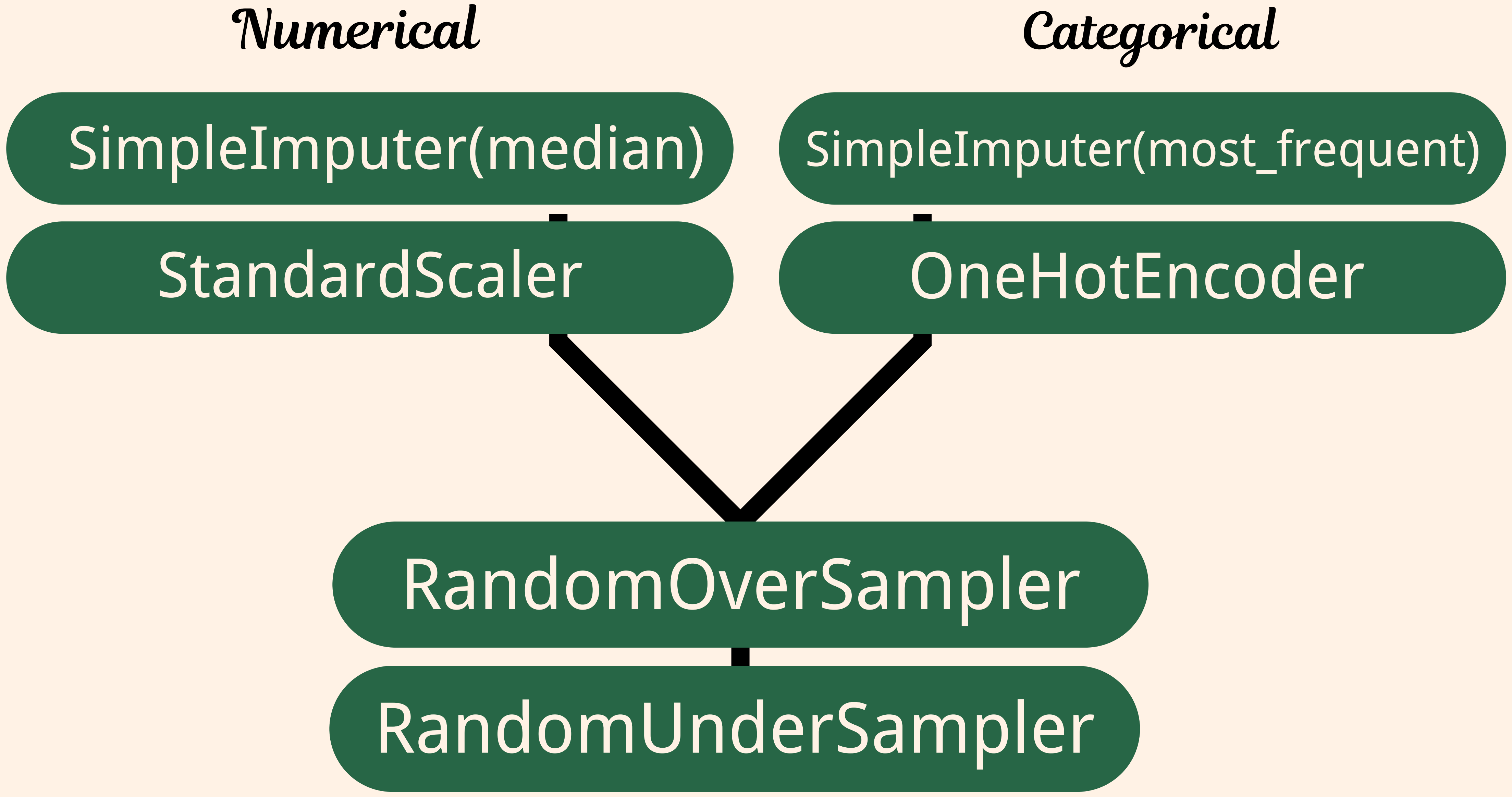
Replace all infinite values from division error

Drop columns that we used for feature extraction



Pipeline construction

- Column Transformer
- Imputer
  - Median for Numerical
  - Mode for Categorical
- Oversample
  - Sampling strategy: 0.1
- Undersample
  - Sampling strategy: majority



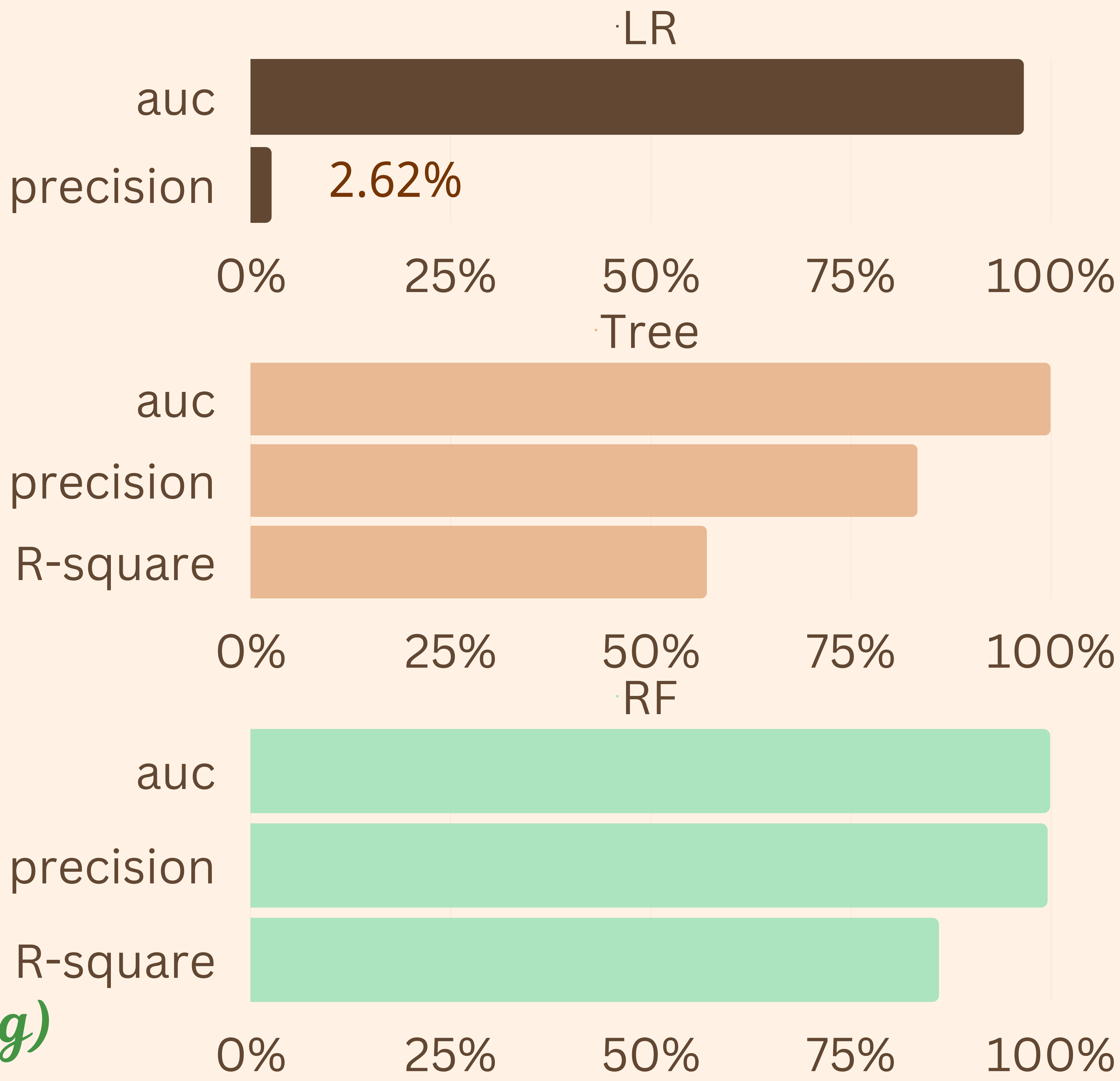
Model Selection

Logistic Regression ✗

Decision Tree ?

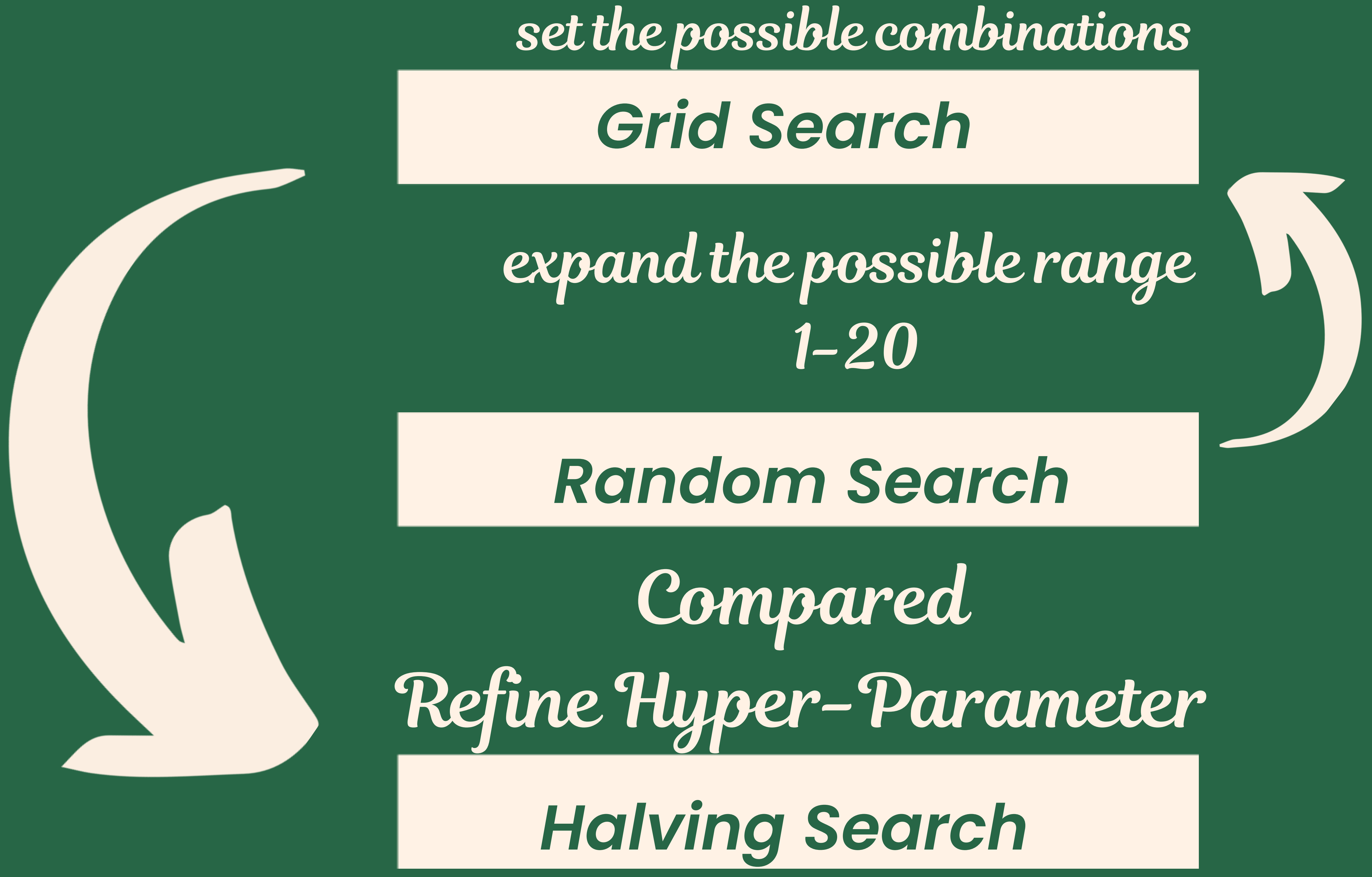
Random Forest

Benchmark score before tuning



(Also, RF has a better performance after tuning)

Hyper-parameter Tuning



set the possible combinations

Grid Search

expand the possible range
1-20

Random Search

Compared
Refine Hyper-Parameter

Halving Search

Decision Tree
Random Forest

	max_depth	min_samples_leaf
Grid	[2, 5, 8]	[2, 5, 7, 8]
Random	randint(1,20)	randint(1,20)
halving	[15, 17,19]	[8,10,11]

Cross-validation

Grid Search

	param_random_forest__max_depth	param_random_forest__min_samples_leaf	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score
5	5	5	0.996716	0.997863	0.998541	0.997707	0.000753
4	5	2	0.996715	0.997863	0.998542	0.997706	0.000754
6	5	7	0.996716	0.997861	0.998542	0.997706	0.000753
7	5	8	0.996712	0.997862	0.998541	0.997705	0.000755
8	8	2	0.996716	0.997866	0.998314	0.997632	0.000673

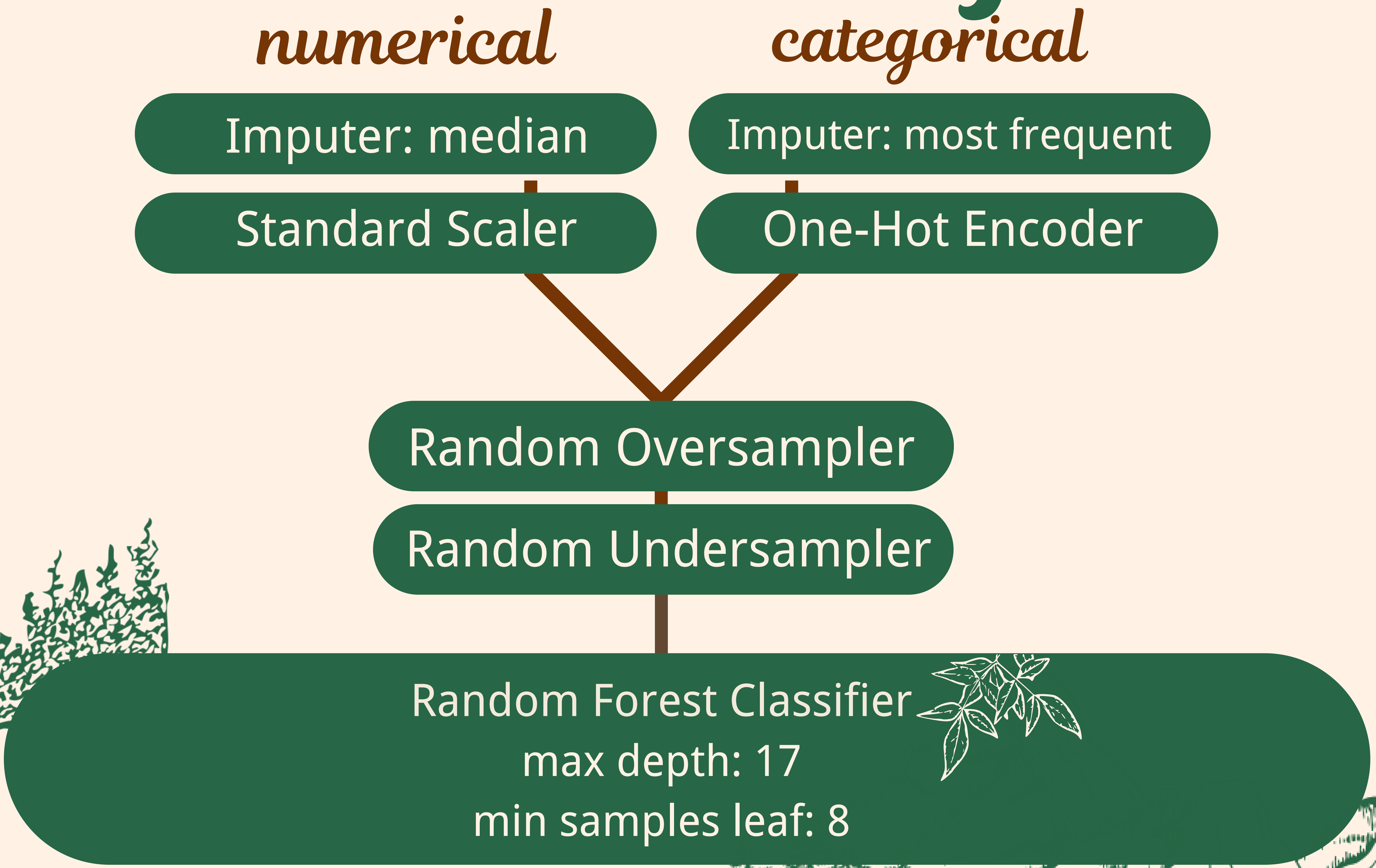
Random Search

	param_random_forest__max_depth	param_random_forest__min_samples_leaf	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score
4	4	8	0.996711	0.998091	0.998538	0.997780	0.000778
1	11	8	0.996716	0.997863	0.998313	0.997631	0.000672
6	12	6	0.996716	0.997862	0.998313	0.997630	0.000672
2	7	19	0.996715	0.997860	0.998314	0.997630	0.000673
3	11	11	0.996716	0.997862	0.998311	0.997630	0.000672

Halving Search

	param_random_forest__max_depth	param_random_forest__min_samples_leaf	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score
12	17	8	0.996715	0.997857	0.998308	0.997627	0.000671
10	17	8	0.997164	0.996560	0.998554	0.997426	0.000835
9	19	11	0.997165	0.996559	0.998554	0.997426	0.000835
11	15	8	0.997164	0.996554	0.998558	0.997425	0.000839
0	15	8	0.995994	0.997890	0.997634	0.997173	0.000840

Final Workflow



numerical

Imputer: median

Standard Scaler

categorical

Imputer: most frequent

One-Hot Encoder

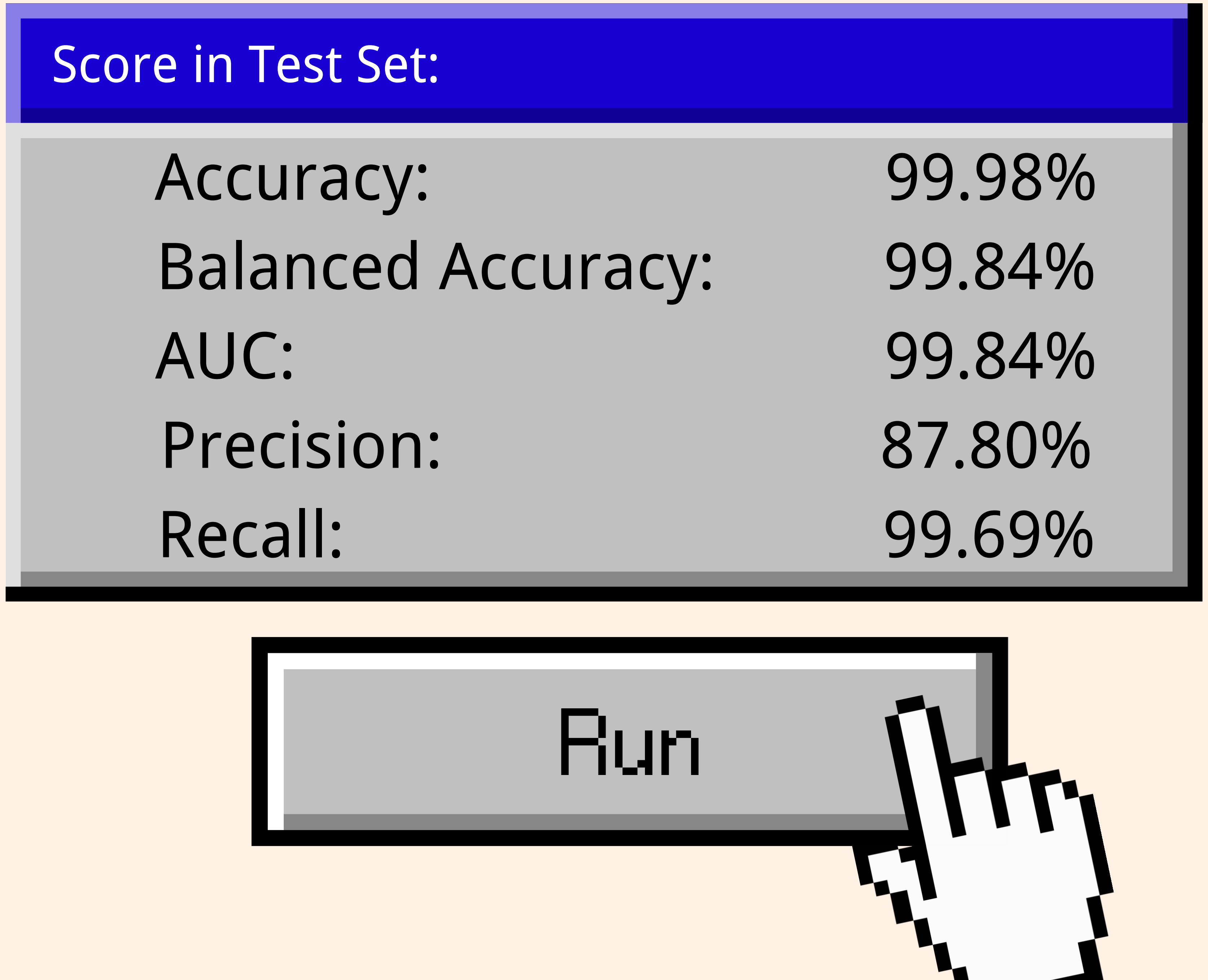
Random Oversampler

Random Undersampler

Random Forest Classifier
max depth: 17
min samples leaf: 8

Performance Summary

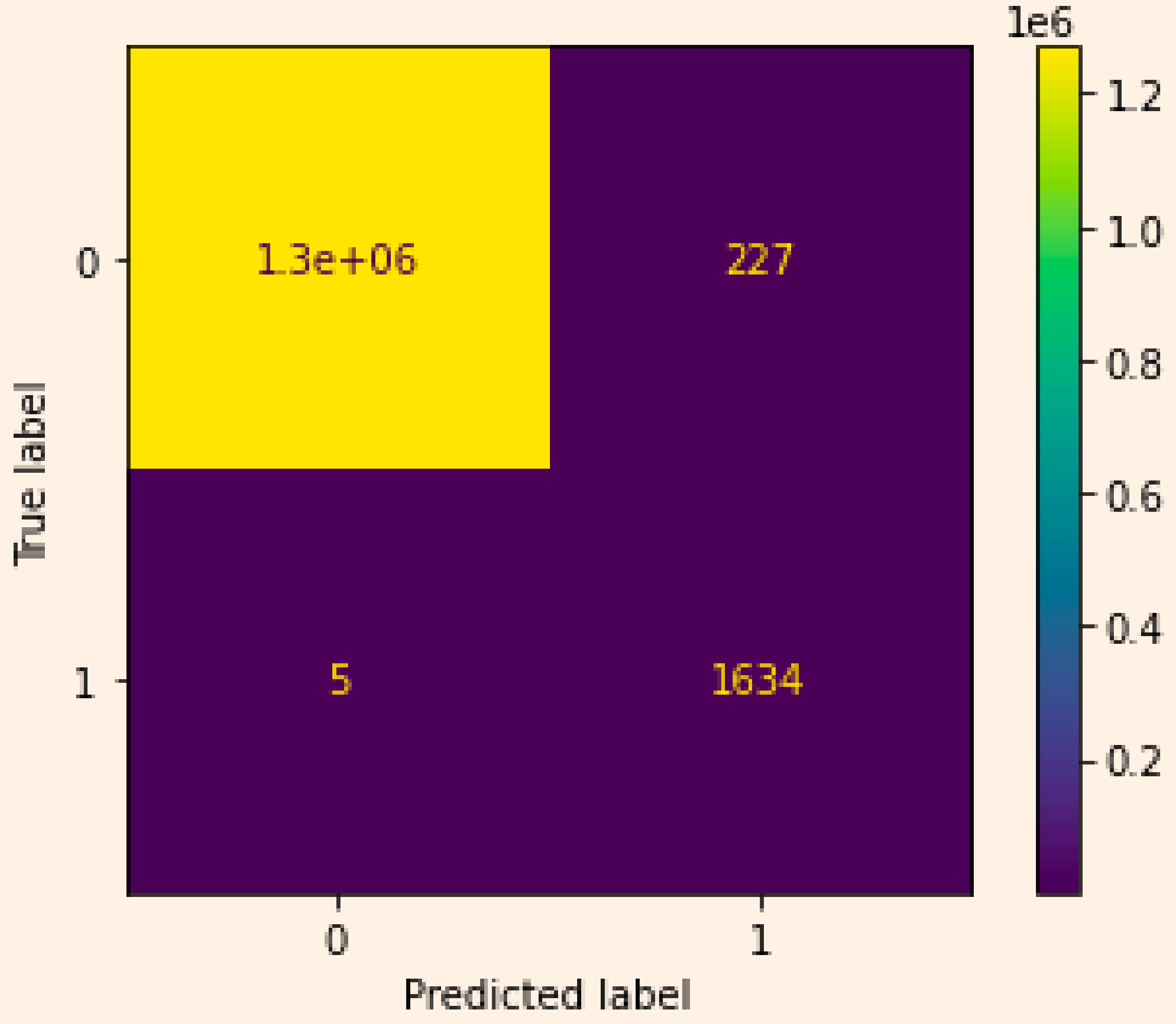
test data



Score in Test Set:

Accuracy:	99.98%
Balanced Accuracy:	99.84%
AUC:	99.84%
Precision:	87.80%
Recall:	99.69%

Confusion Matrix



Closing Remarks

Unresolved Challenges

Cost matrix
Feature engineering within pipeline
Runtime optimization & parameter search

Summary

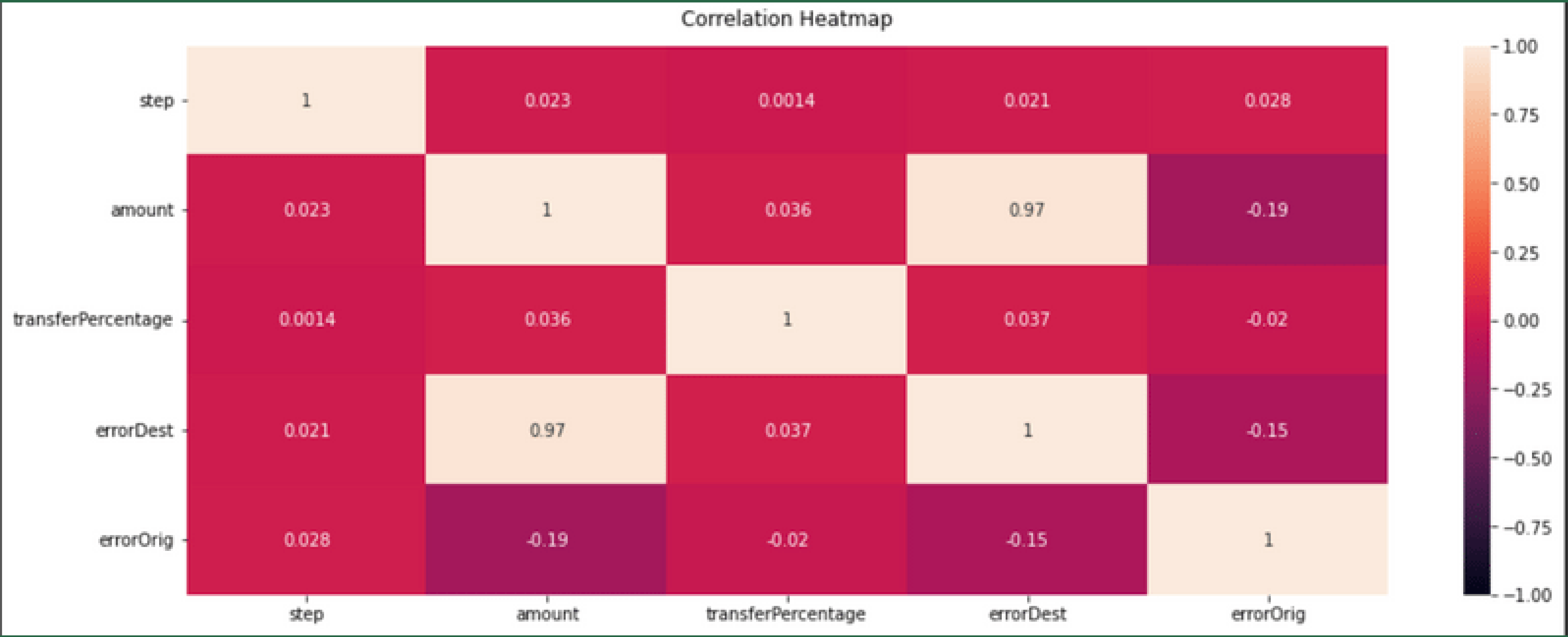
Model Selection & Best Parameter
Metric Evaluation

Conclusion

> “The implemented machine-learning model effectively identifies fraud within the dataset, and will reduce the risk of fraud and efficiently advise customers of potential fraud. However, the column limitation can restrict its broader impact.”

Notebook link: https://colab.research.google.com/drive/1fk3YjF9j8xl893SzFtUWxx75jpKv8y6C?usp=sharing

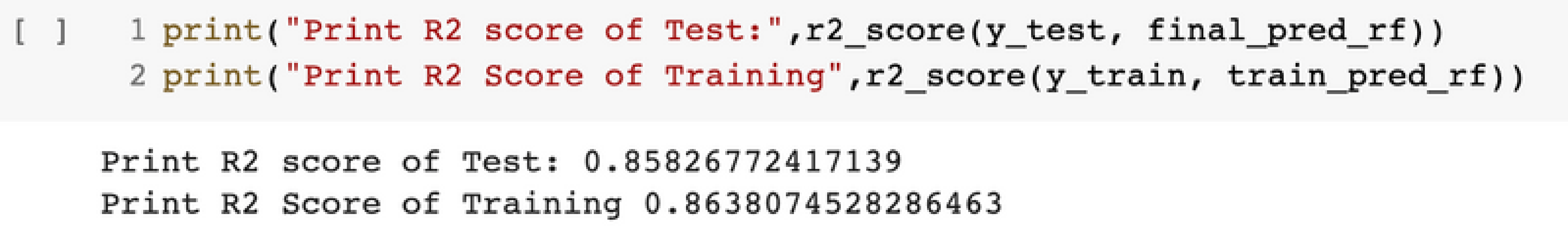
Thank You!



Correlation Heatmap

	step	amount	transferPercentage	errorDest	errorOrig
step	1	0.023	0.0014	0.021	0.028
amount	0.023	1	0.036	0.97	-0.19
transferPercentage	0.0014	0.036	1	0.037	-0.02
errorDest	0.021	0.97	0.037	1	-0.15
errorOrig	0.028	-0.19	-0.02	-0.15	1

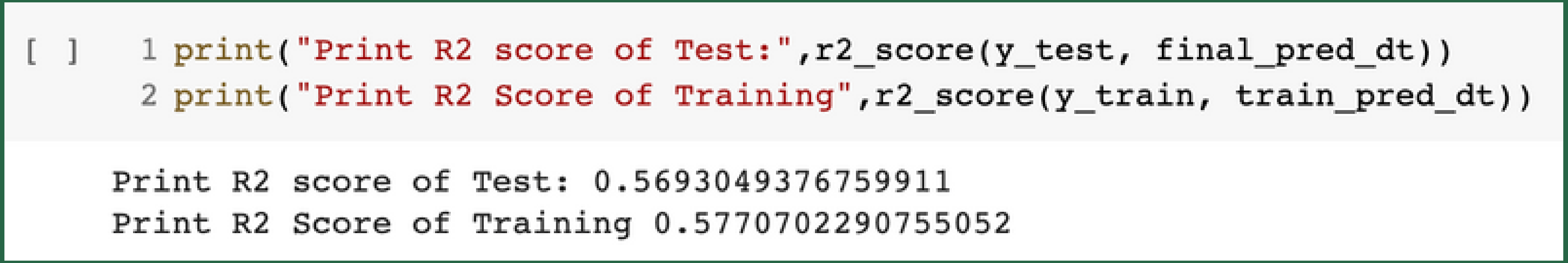
RandomForest:



```python
print("Print R2 score of Test:",r2_score(y_test, final_pred_rf))
print("Print R2 Score of Training",r2_score(y_train, train_pred_rf))
```

```
Print R2 score of Test: 0.85826772417139
Print R2 Score of Training 0.8638074528286463
```

DecisionTree:



```python
print("Print R2 score of Test:",r2_score(y_test, final_pred_dt))
print("Print R2 Score of Training",r2_score(y_train, train_pred_dt))
```

```
Print R2 score of Test: 0.5693049376759911
Print R2 Score of Training 0.5770702290755052
```