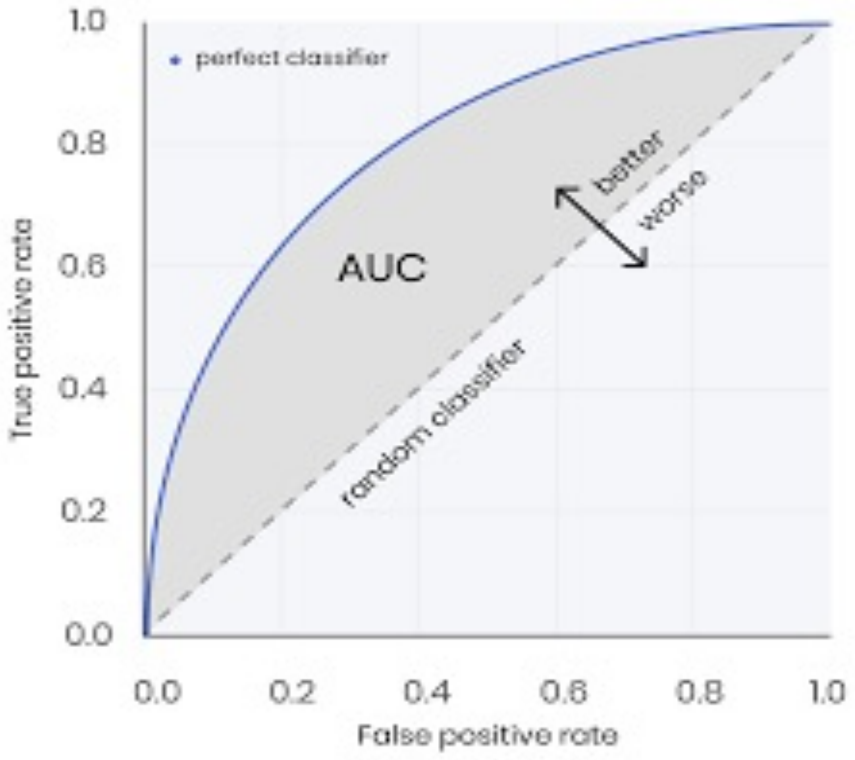
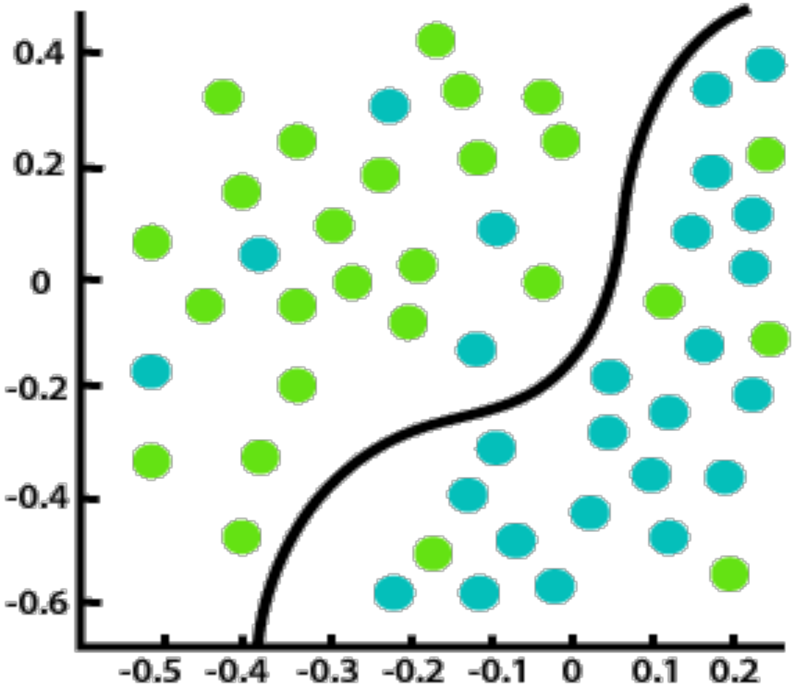
Project for STA6247.

Classification of default event(loan).





Antasiuk Vladimir

The problem: classification of non-default event (loan).

Default vs non-default events (loans).

The aim is to identify patterns which indicate if a person is likely to default, which may be used for taking actions such as denying the loan, reducing the amount of loan, lending (to risky applicants) at a higher interest rate, etc.

<u>Two types of risks are associated with the bank’s decision:</u>

- If the applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company
- If the applicant is not likely to repay the loan, i.e. he/she is likely to default, then approving the loan may lead to a financial loss for the company

The data

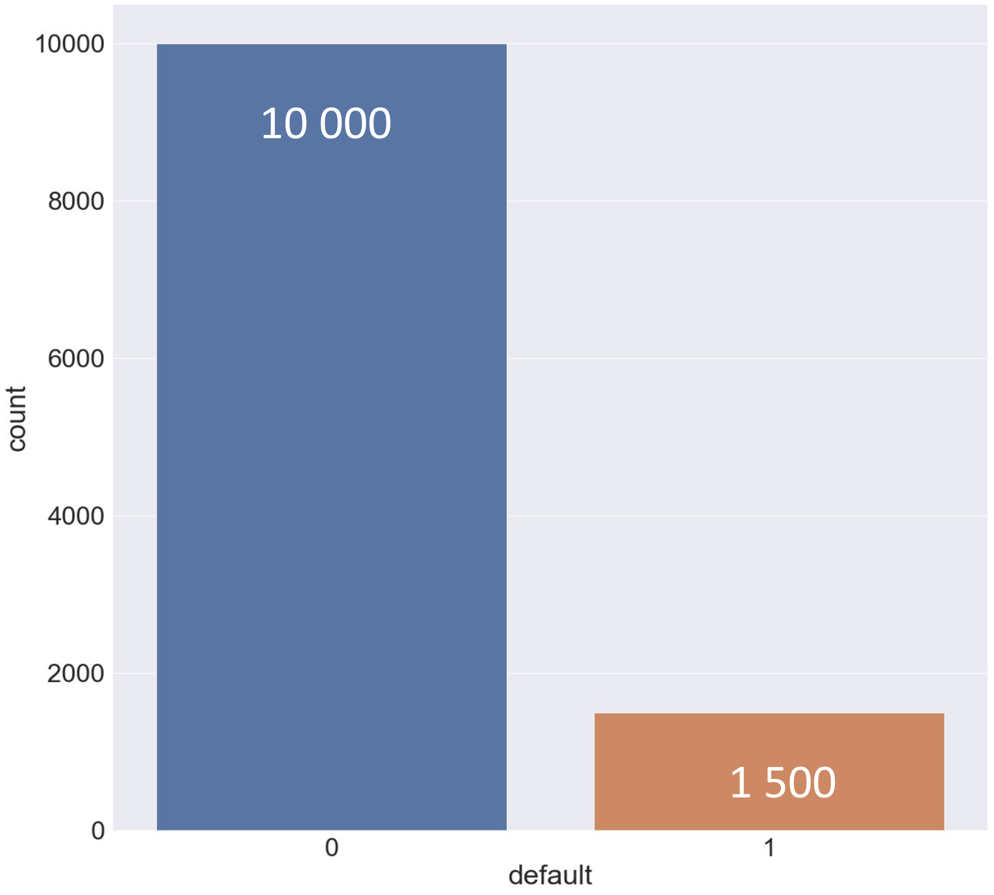
Data obtained from a private company

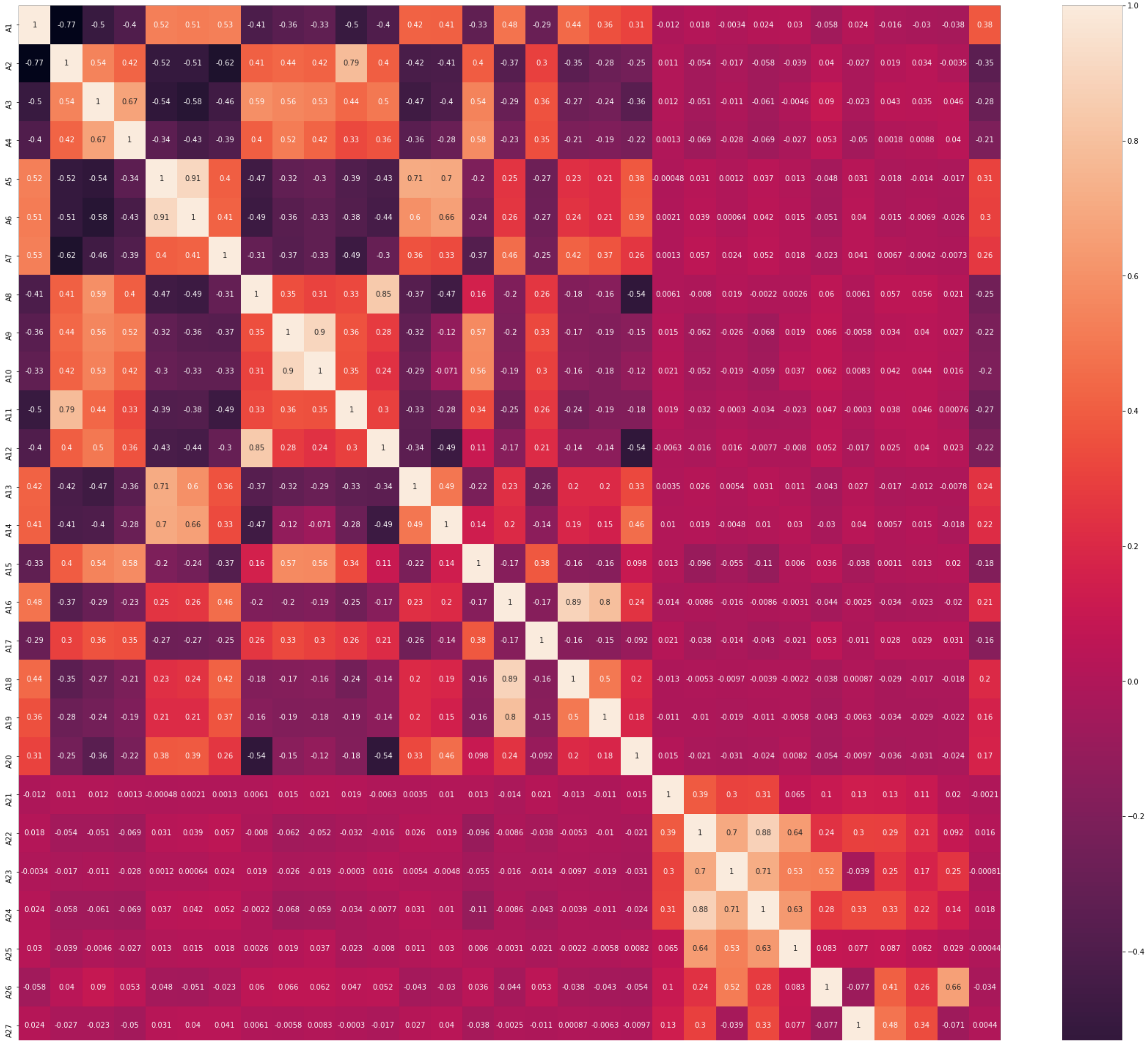
30 encrypted features have encoded names like A1, A2 … but we don’t know what does it really mean)

Train and test sets the same size (11500 rows, 30 features)

Target class: default=1, non-default=0. 13% of defaults in each dataset

Distribution of classes in train and test datasets

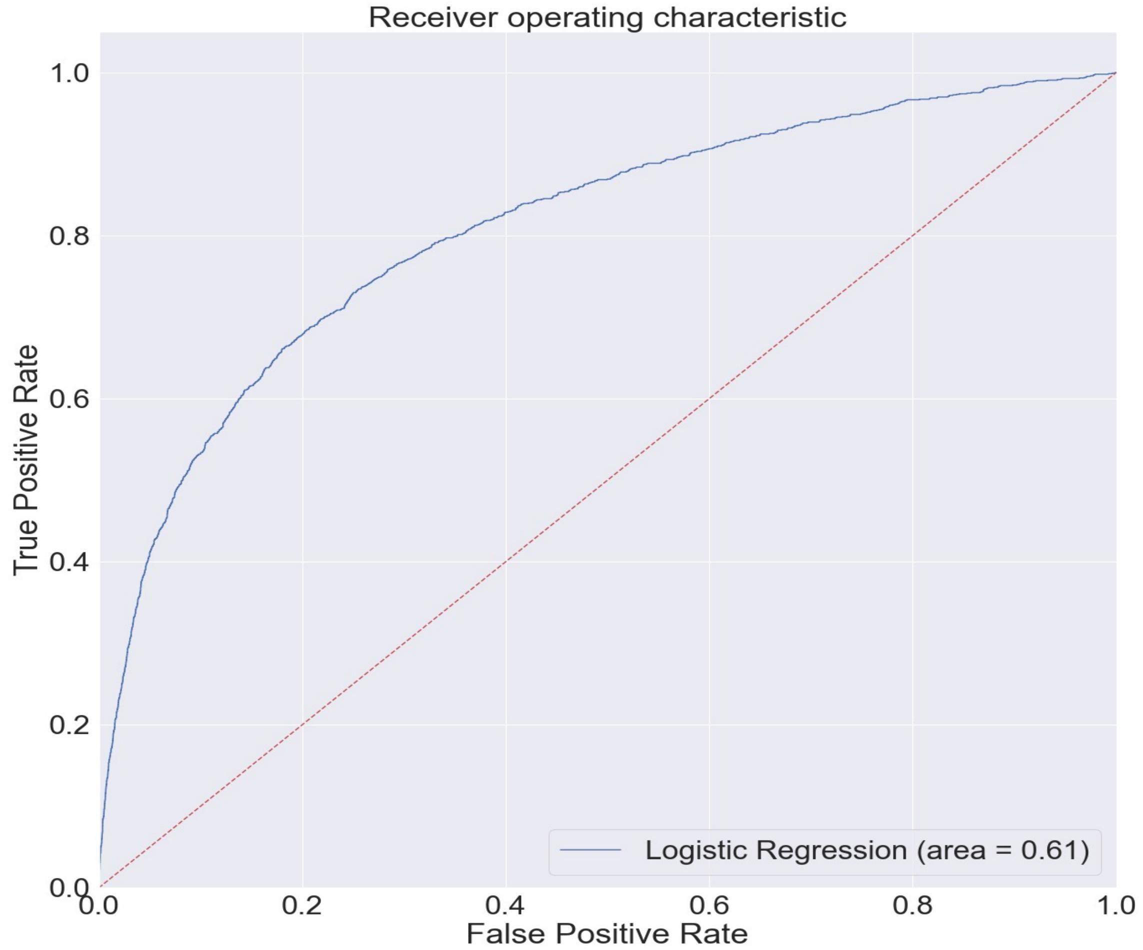




Finding and excluding highly correlated features

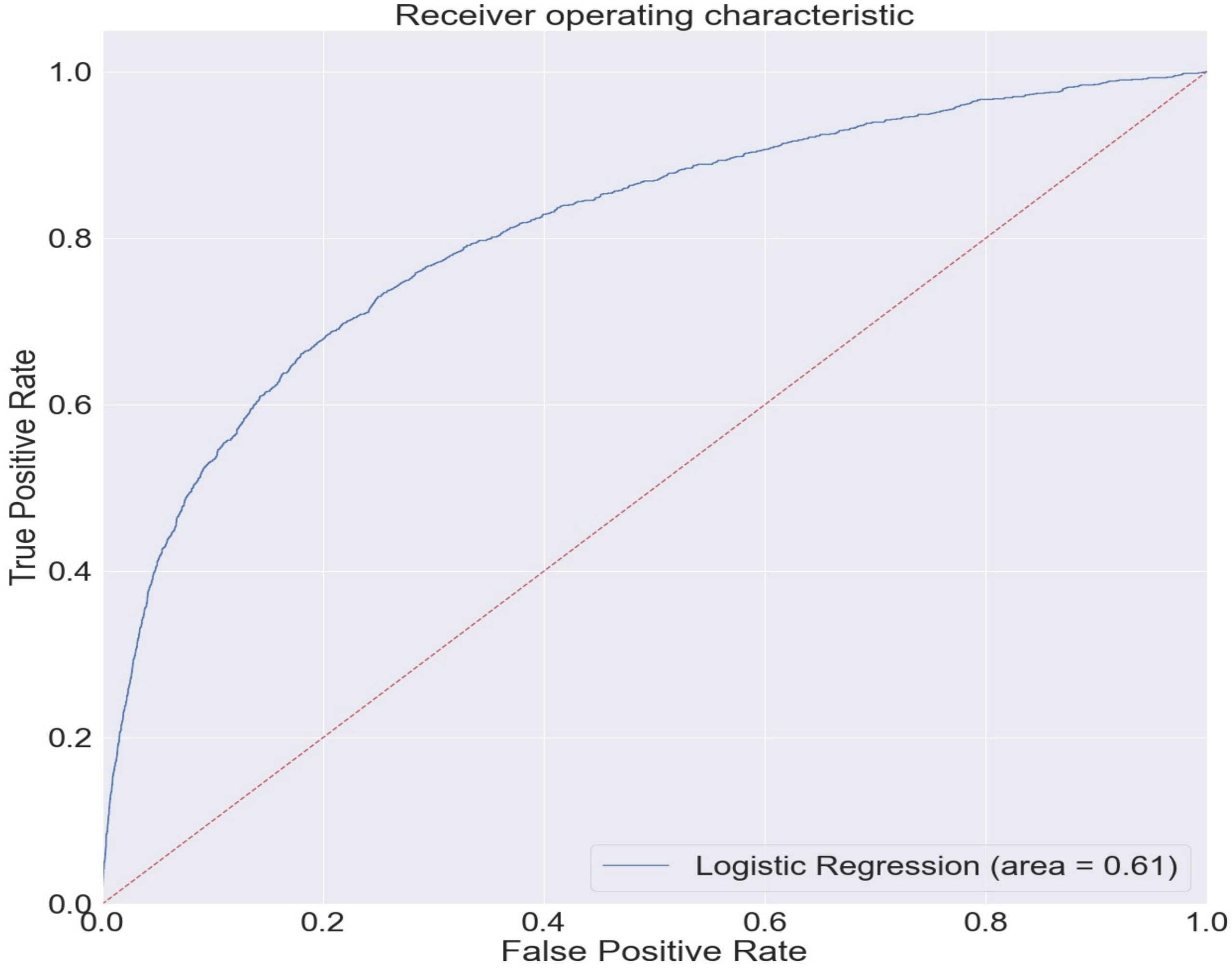
Logistic Regression – 10 features (train set)

	precision	recall	f1-score	support
0	0.90	0.98	0.94	10000
1	0.63	0.23	0.34	1500
accuracy			0.88	11500
macro avg	0.76	0.61	0.64	11500
weighted avg	0.86	0.88	0.86	11500



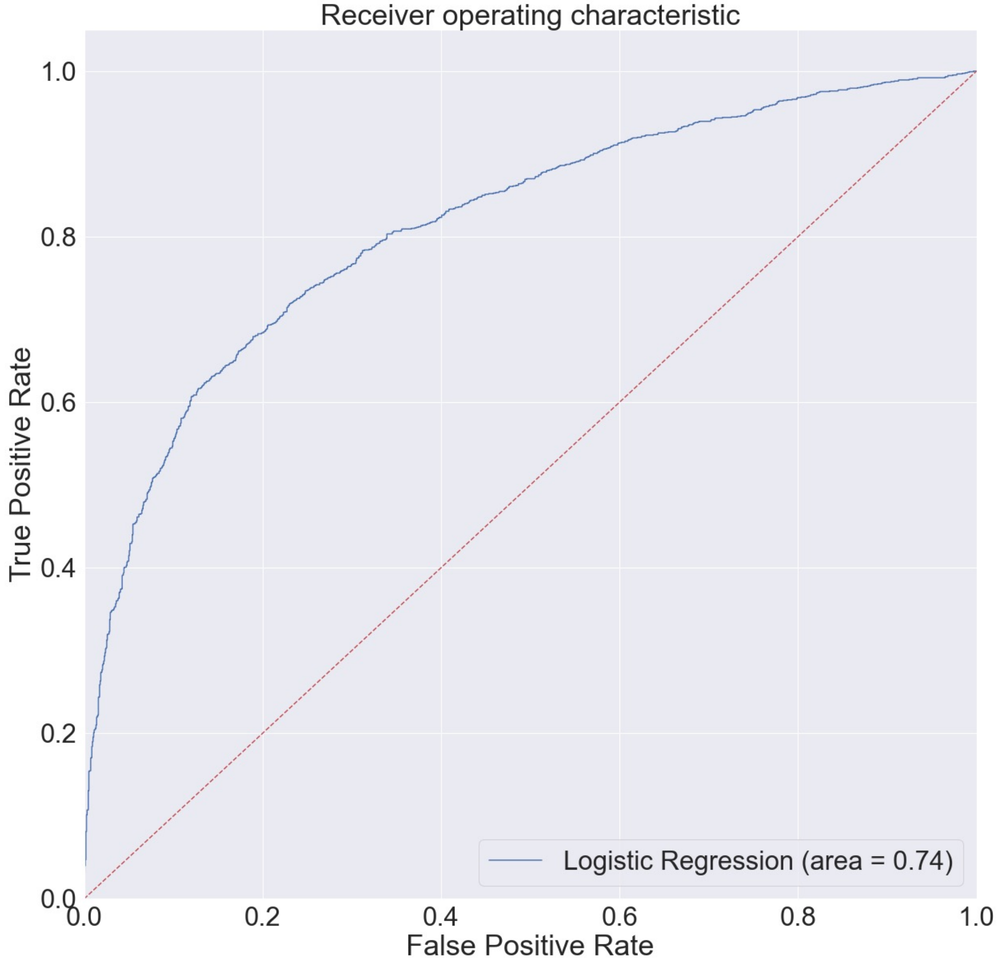
Logistic Regression – 10 features (test set)

	precision	recall	f1-score	support
0	0.90	0.98	0.94	10000
1	0.63	0.23	0.34	1500
accuracy			0.88	11500
macro avg	0.76	0.61	0.64	11500
weighted avg	0.86	0.88	0.86	11500



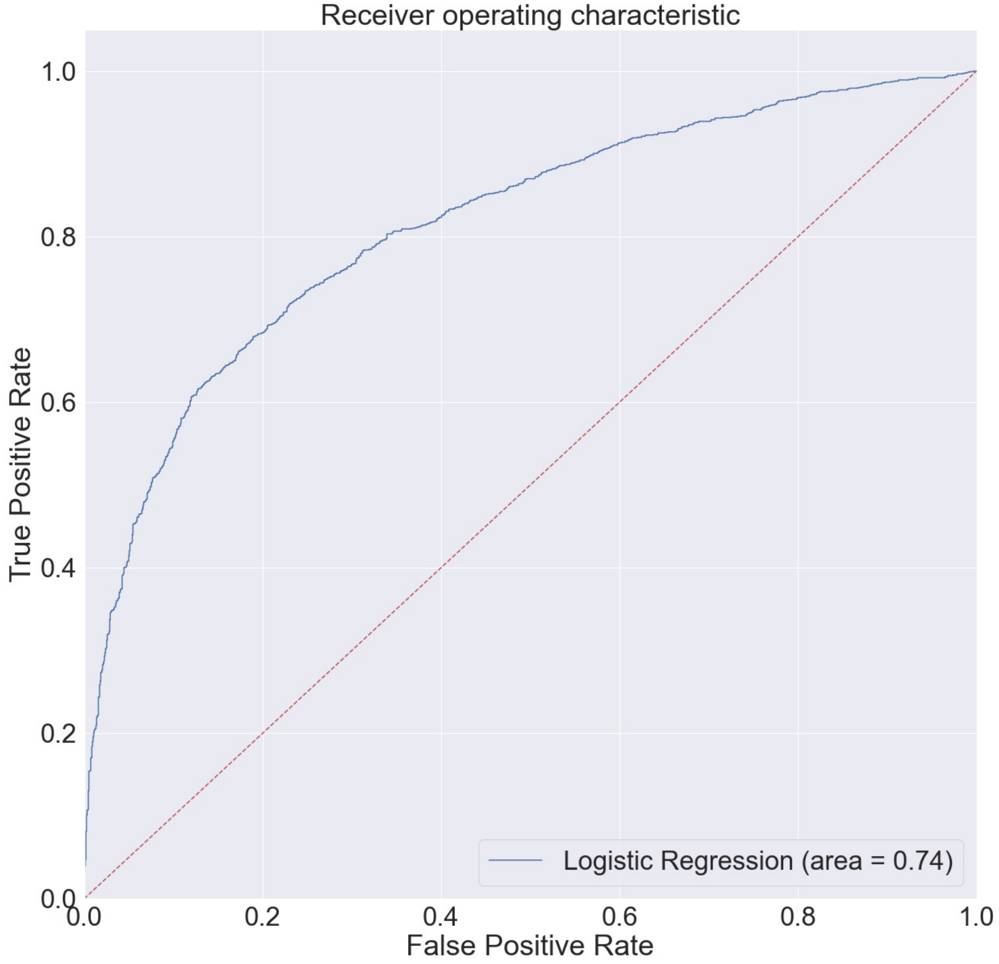
Logistic Regression (undersampling, train set). 7 variables

	precision	recall	f1-score	support
0	0.74	0.74	0.74	1500
1	0.74	0.74	0.74	1500
accuracy			0.74	3000
macro avg	0.74	0.74	0.74	3000
weighted avg	0.74	0.74	0.74	3000



Logistic Regression (undersampling, test set). 7 variables.

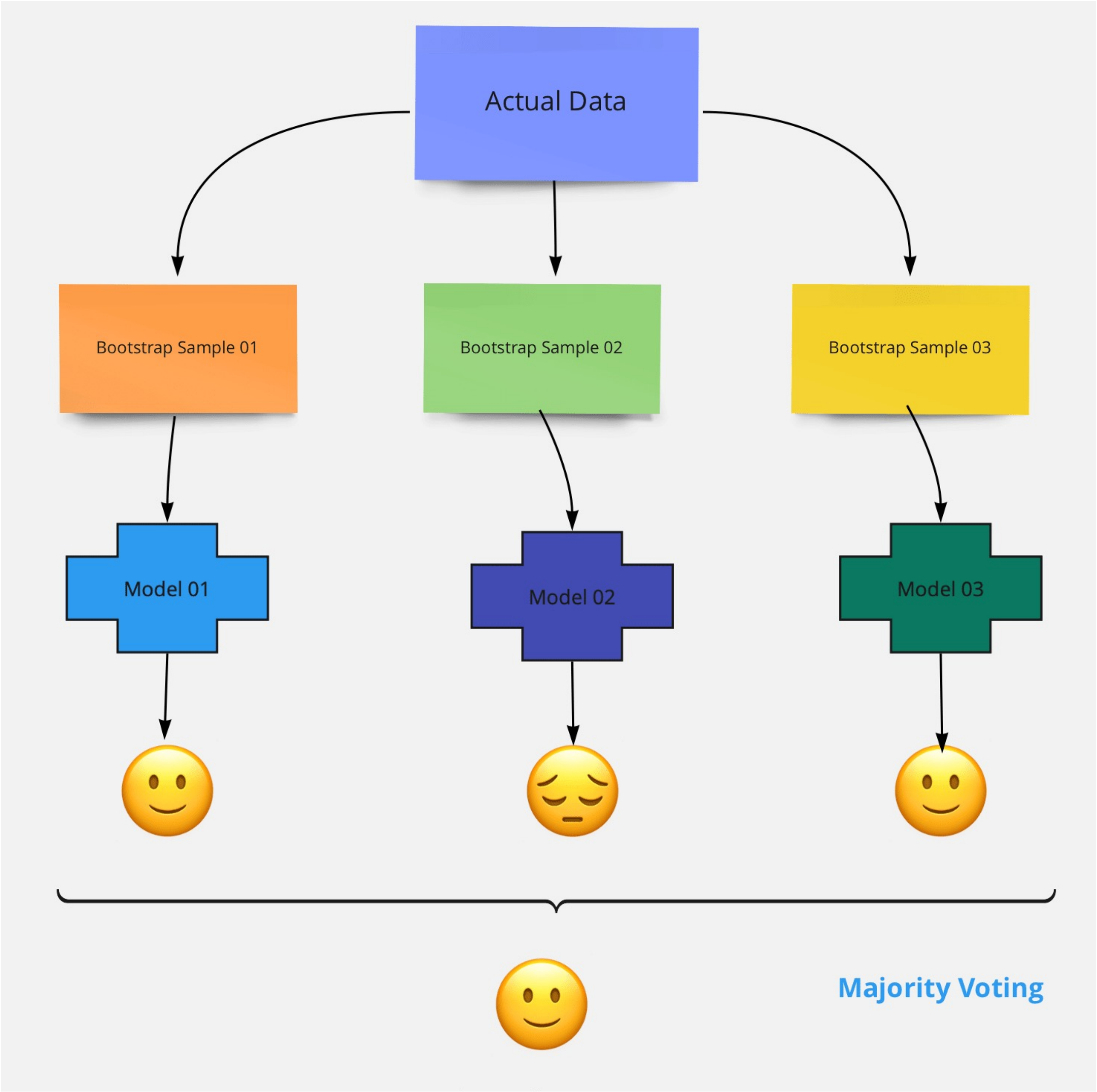
	precision	recall	f1-score	support
0	0.95	0.75	0.84	10000
1	0.31	0.74	0.43	1500
accuracy			0.75	11500
macro avg	0.63	0.74	0.64	11500
weighted avg	0.87	0.75	0.79	11500



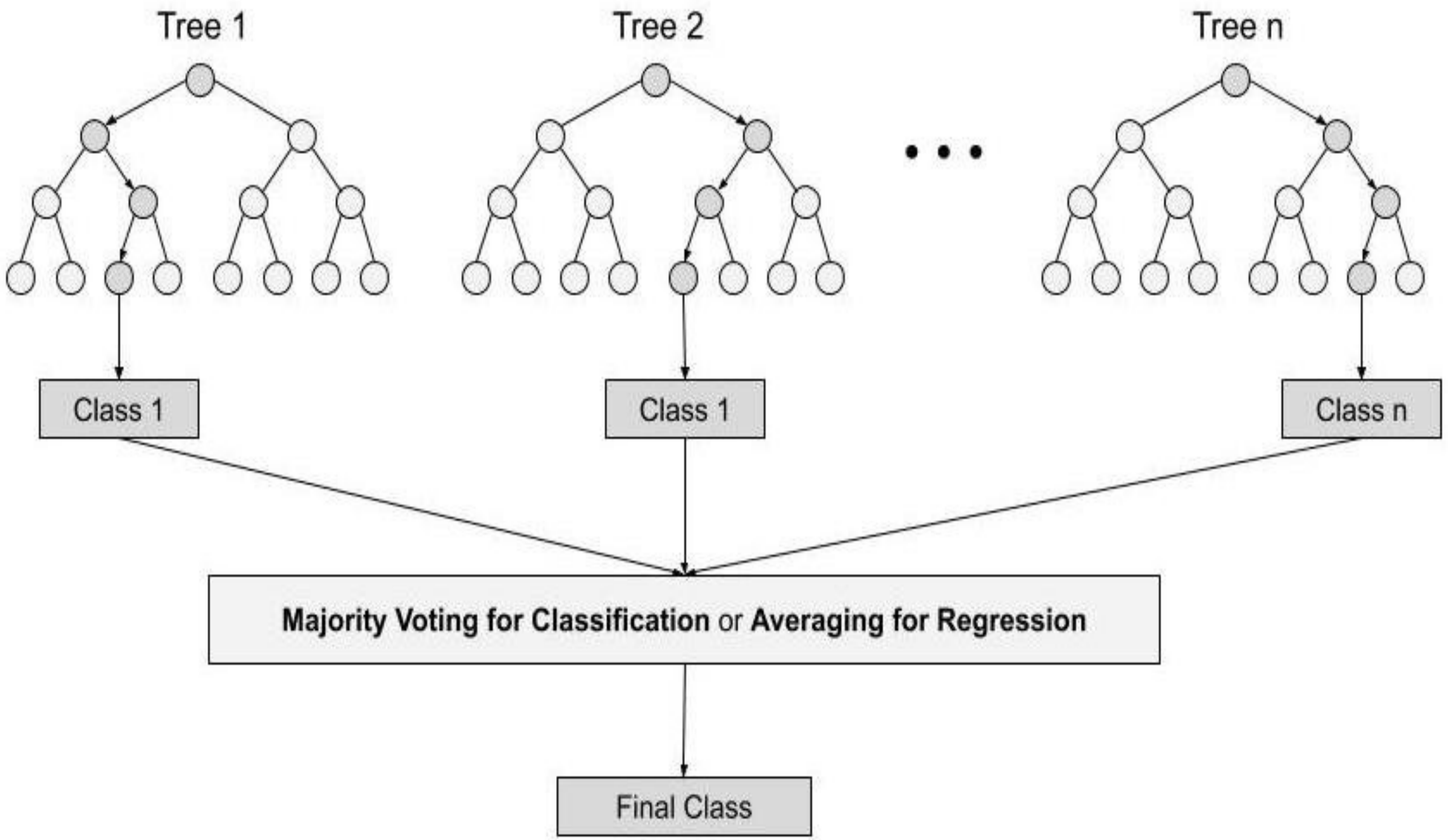
Random Forest Classifier

- <u>Step 1</u>: In Random forest n number of random records are taken from the data set having k number of records.
- <u>Step 2</u>: Individual decision trees are constructed for each sample.
- <u>Step 3</u>: Each decision tree will generate an output.
- <u>Step 4</u>: Final output is considered based on ***Majority Voting or Averaging*** for Classification and regression respectively.

Bagging Ensemble Method

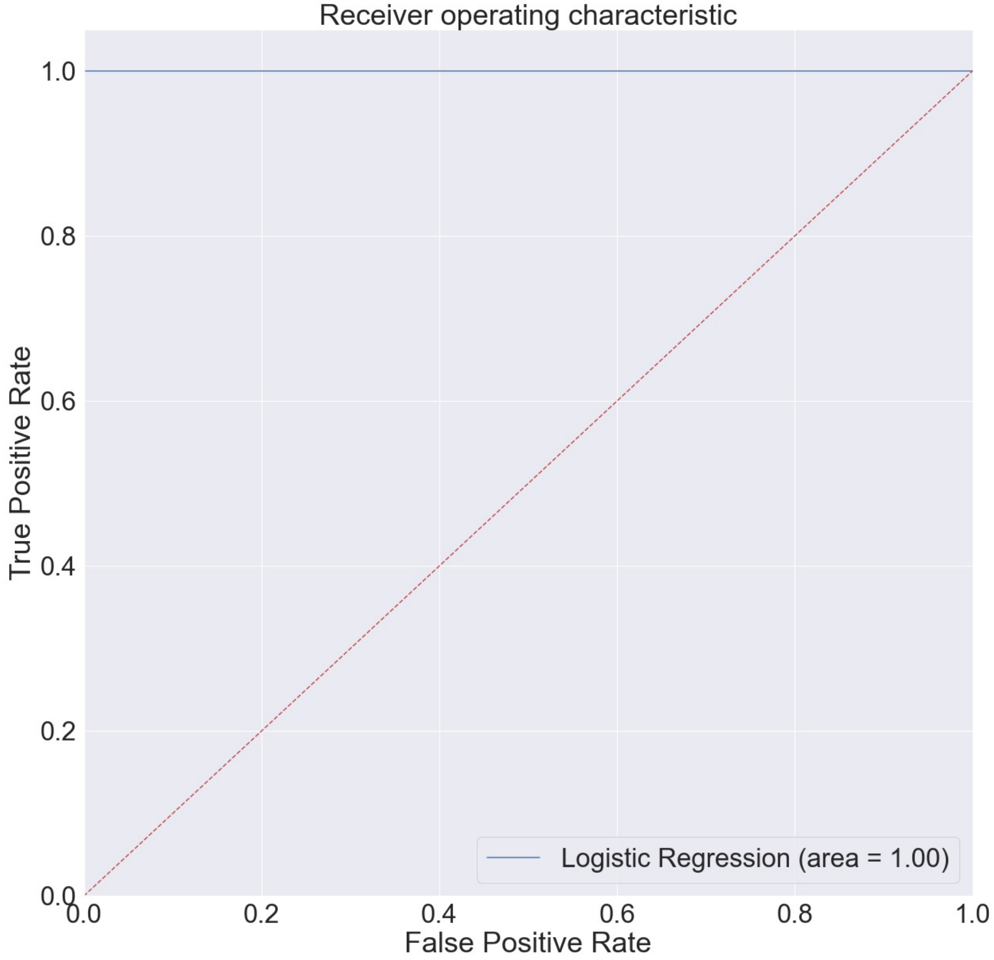


Random Forest Classifier



Random Forest Classifier (same confusion matrix and AUC on train and test sets)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10000
1	1.00	1.00	1.00	1500
accuracy			1.00	11500
macro avg	1.00	1.00	1.00	11500
weighted avg	1.00	1.00	1.00	11500



Conclusion

- We don’t need all 30 variables to build sufficient logistic regression model
- Undersampling/oversampling can help improve model precision (in our case AUC improved from 0.61 to 0.74 with lighter model)
- Sophisticated algorithms can improve model quality (Random forest classifier gave us perfect AUC)