CS699- Data Mining

TERM PROJECT REPORT

Practice of building and testing classification models using a real-world dataset.

SUBMITTED TO: Prof. J. Young Lee

SUBMITTED BY:
SARTHAK MAHAPATRA (#U88495871)
NAHUSH BHAMRE (#U41018088))

1. PREPROCESSING STEPS

In our intermediate report, beginning with the initial dataset containing 5000 observations across 276 columns, we categorized features into numerical and categorical variables, resulting in 240 categorical and 26 numerical columns. Through a series of preprocessing steps tailored for each variable type, we addressed issues such as unbalanced distributions, missing values, and multicollinearity.

For categorical columns, we removed unbalanced columns, handled missing values, and eliminated correlated columns, resulting in a refined set of 50 categorical features. Similarly, for numerical columns, we employed outlier removal techniques, imputed missing values, and removed correlated columns, resulting in a refined set of 8 numerical features.

Subsequently, in the combined data frame, we further addressed multicollinearity issues, removed outliers, and applied min-max scaling to standardize the numerical data.

Overall, we have 3704 observations across 57 columns, and these preprocessing steps have laid a solid foundation for subsequent analysis and modelling tasks. By optimizing the dataset's quality and suitability, we are poised to develop a robust classification model that effectively addresses the project's objectives.

2. DATASET BALANCING METHODS

Oversampling, and undersampling are the two techniques that are used in data preprocessing for addressing class imbalance in our datasets.

2.1. Oversampling:

Oversampling involves increasing the number of instances in the minority class to balance it with the majority class. This can be achieved through various techniques such as duplication of existing instances (e.g., replication), generating synthetic instances using algorithms like Synthetic Minority Over-sampling Technique (SMOTE), or creating new instances through interpolation methods. Oversampling helps prevent the classifier from being biased towards the majority class and improves its ability to learn from the minority class, thus enhancing the overall predictive performance, especially in scenarios where the minority class is of significant interest.

In our implementation, we first subset the training data to extract instances belonging to the minority class . Then, a random sample of size 2000 (adjustable) is drawn with replacement from the indices of these instances. These sampled instances are appended to the original dataset, effectively oversampling the minority class.

2.2. Undersampling:

Undersampling aims to reduce the number of instances in the majority class to balance the class distribution. This can be done by randomly removing instances from the majority class until a more balanced distribution is achieved. Undersampling helps alleviate the computational burden associated with imbalanced datasets and can also mitigate the issue of overfitting to the majority class. However, undersampling may discard potentially valuable information present in the majority class, leading to loss of predictive power, especially when the majority class contains crucial patterns or insights.
Similarly, undersampling is implemented by first subsetting the training data to extract instances belonging to the majority class. Then, a random sample of size is drawn without replacement from the indices of these instances. These sampled instances are appended to the original dataset, effectively undersampling the majority class.

3. FEATURE SELECTION METHODS:

3.1. Correlation-based Feature Selection (CFS):

CFS is a feature selection method that evaluates the worth of a subset of features by considering the individual feature evaluation and the redundancy among them. It aims to select features that are highly correlated with the class while being uncorrelated with each other.
To implement CFS, using R we calculate the correlation between each feature and the target class, as well as the correlation between features themselves. Then, a subset of features is selected based on these correlations to maximize the relevance to the class while minimizing redundancy.

3.2. Boruta:

Boruta is a feature selection algorithm that works as a wrapper built around a random forest classifier. It identifies important features by comparing the importance of real features with the importance of randomly shuffled features. Boruta iteratively conducts feature selection until it determines the significance of each feature with respect to the target class.
The Boruta algorithm is implemented using the 'Boruta' package in R. It utilizes the Boruta function, which takes the formula representing the target class and features as input. Boruta then iteratively evaluates feature importance and selects relevant features.

3.3 Information Gain:

Information Gain is a feature selection method commonly used in decision trees and other tree-based models. It measures the reduction in entropy or uncertainty about the target class provided by the knowledge of the presence of a particular feature. Features with higher information gain are considered more important for classification.
For our implementation, Information Gain is implemented using the 'FSelector' package in R. The 'information.gain' function calculates the information gain of each feature with respect to the target class. Features are then ranked based on their information gain values, and a subset of the 40 most informative features is selected for further analysis.

4. MACHINE LEARNING ALGORITHMS USED:

4.1. Adaboost (Adaptive Boosting):

Adaboost is an ensemble learning technique that sequentially trains weak learners, focusing on instances that were misclassified by previous learners. It assigns higher weights to misclassified instances, thereby allowing subsequent learners to focus more on difficult cases. The 'AdaBoost.M1' method in the code implements Adaboost, training a series of weak learners and combining their predictions to produce a strong classifier.

4.2. Bagged Flexible Discriminant Analysis:

Bagged FDA enhances the performance of Flexible Discriminant Analysis by leveraging bootstrap aggregation (bagging) to improve stability and accuracy. It builds multiple Flexible Discriminant Analysis models on bootstrap samples of the dataset and aggregates their predictions to reduce variance and overfitting. The 'bagFDA' method in the code implements this approach by training multiple Flexible Discriminant Analysis models on bootstrap samples and combining their predictions**.**

4.3. Conditional Inference Random Forest:

Conditional Inference Random Forest is an extension of Random Forest that incorporates statistical tests to make unbiased variable selection and splitting decisions. It uses permutation tests or other statistical measures to determine the significance of feature splits, reducing the potential for overfitting. In the provided code, the 'cforest' method implements this approach by training a Conditional Inference Random Forest model with specified parameters.

4.4. Flexible Discriminant Analysis (FDA):

FDA is a classification method that generalizes standard linear discriminant analysis (LDA) by allowing for non-linear decision boundaries. It estimates class probabilities by modeling the distribution of features within each class and applies flexible techniques to accommodate non-linear relationships. In the provided code, the 'fda' method is used for classification tasks, where FDA is employed to model the relationship between predictors and the target class.

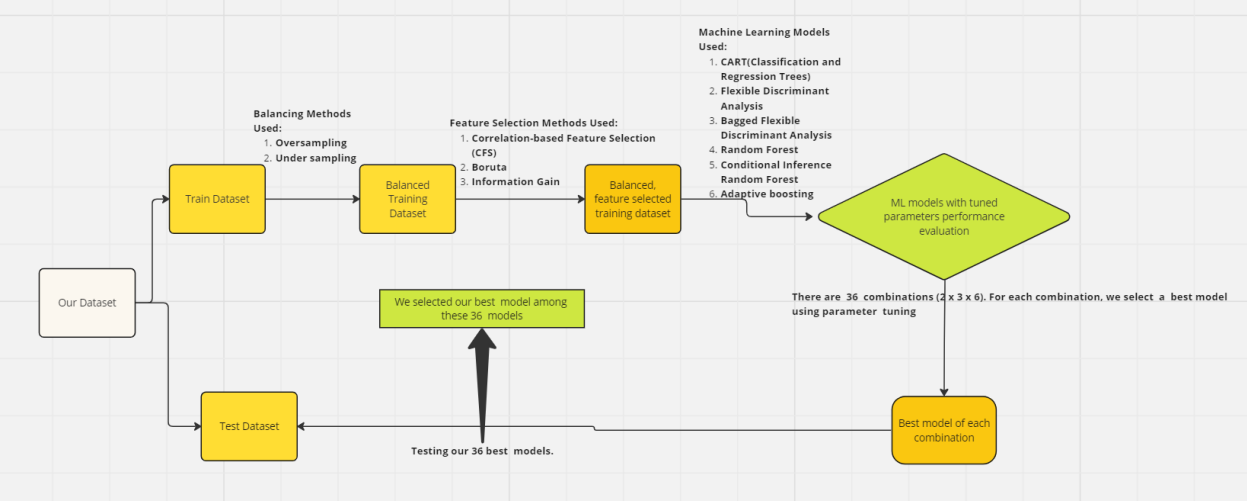
4.5. Random Forest:

Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of individual trees. It introduces randomness in tree construction by selecting a random subset of features for each tree and using bootstrapped samples for training. The 'rf' method in the code implements Random Forests, where a forest of decision trees is trained and aggregated to make predictions.

4.6. CART (Classification and Regression Trees):

CART is a decision tree algorithm used for both classification and regression tasks. It recursively splits the dataset into subsets based on the value of features, making decisions at each node to minimize impurity or maximize information gain. This process creates a tree structure where each leaf node represents a class or a regression value. The 'rpart' implementation in the provided code utilizes recursive partitioning for building decision trees. It selects the optimal splits based on measures like Gini index or information gain, specified through parameters.

5. OUR DATA MINING PROCEDURE:



We have made used nested for loops using R Code which performs a series of experiments involving all our data preprocessing, feature selection, model training, and evaluation. We have broken down our workflow and explained each part:

5.1. Data Splitting:

The dataset is split into training and testing sets using the `initial_split` function from the `rsample` package. The training set contains approximately 66.7% of the data, and the testing set contains the remaining data.

5.2. Balancing and Feature Selection:

We have enlisted all the balancing techniques (`balancings`) and feature selection methods (`feature_selections`). It then iterates through each combination of balancing and feature selection methods.

5.3. Model Training and Evaluation:

Inside the nested loops, each combination of balancing, feature selection, and ML model is trained and evaluated. For each combination, the training data is balanced using the specified technique, features are selected, and then the model is trained using the selected features. Predictions are made on the test set, and the actual results are compared to evaluate the model's performance.

5.4. Results Storage:

For each combination, the evaluation results are stored in a CSV file named based on the combination of balancing, feature selection, and ML model used.

6. Results and Observation:

The results have been mentioned in the format:
(Balancing method) + (Feature selection method) + (Classification Algorithm)

- Oversampling + Boruta + AdaBoost

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1674    |         344         |
| Negative |     911     |         1088        |
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.8294	0.4558	0.5442	0.1706	0.8903	0.8294	0.8588	0.6868	0.3388	0.3341
Class 1	0.5442	0.1706	0.8294	0.4558	0.4169	0.5442	0.4721	0.6868	0.3388	0.3341
Wt. Average	0.7772	0.4036	0.5964	0.2228	0.8036	0.7772	0.788	0.6868	0.3388	0.3341

- Oversampling + Boruta + bagFDA

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1548    |         470         |
| Negative |     699     |         1300        |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7669	0.3496	0.6504	0.2331	0.9073	0.7669	0.8312	0.7087	0.3491	0.3292
Class 1	0.6504	0.2331	0.7669	0.3496	0.3848	0.6504	0.4835	0.7087	0.3491	0.3292
Wt. Average	0.7456	0.3283	0.6717	0.2544	0.8116	0.7456	0.7675	0.7087	0.3491	0.3292

- Oversampling + Boruta + cforest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1770    |         248         |
| Negative |     1106    |         893         |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.877	0.5531	0.4469	0.123	0.8761	0.877	0.8765	0.6619	0.3244	0.3244
Class 1	0.4469	0.123	0.877	0.5531	0.4489	0.4469	0.4479	0.6619	0.3244	0.3244
Wt. Average	0.7982	0.4743	0.5257	0.2018	0.7979	0.7982	0.798	0.6619	0.3244	0.3244

- Oversampling + Boruta + fda

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1526    |         492         |
| Negative |     654     |         1345        |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.756	0.3274	0.6726	0.244	0.9115	0.756	0.8265	0.7143	0.3546	0.3308
Class 1	0.6726	0.244	0.756	0.3274	0.3819	0.6726	0.4872	0.7143	0.3546	0.3308
Wt. Average	0.7407	0.3121	0.6879	0.2593	0.8145	0.7407	0.7644	0.7143	0.3546	0.3308

- Oversampling + Boruta + random forest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1956    |         62          |
| Negative |     1460    |         539         |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.9692	0.7301	0.2699	0.0308	0.8555	0.9692	0.9088	0.6196	0.3522	0.3106
Class 1	0.2699	0.0308	0.9692	0.7301	0.663	0.2699	0.3836	0.6196	0.3522	0.3106
Wt. Average	0.8411	0.602	0.398	0.1589	0.8202	0.8411	0.8126	0.6196	0.3522	0.3106

- Oversampling + Boruta + CART

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1492    |         526         |
| Negative |     876     |         1123        |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7391	0.4381	0.5619	0.2609	0.8827	0.7391	0.8045	0.6505	0.2504	0.2349
Class 1	0.5619	0.2609	0.7391	0.4381	0.3256	0.5619	0.4123	0.6505	0.2504	0.2349
Wt. Average	0.7066	0.4056	0.5944	0.2934	0.7807	0.7066	0.7327	0.6505	0.2504	0.2349

- Oversampling + cfs + AdaBoost

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1548    |         470         |
| Negative |     893     |         1106        |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7669	0.4469	0.5531	0.2331	0.8844	0.7669	0.8215	0.66	0.2723	0.2601
Class 1	0.5531	0.2331	0.7669	0.4469	0.3472	0.5531	0.4266	0.66	0.2723	0.2601
Wt. Average	0.7277	0.4077	0.5923	0.2723	0.786	0.7277	0.7492	0.66	0.2723	0.2601

- Oversampling + cfs + bagFDA

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1554    |         464         |
| Negative |     770     |         1230        |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7698	0.385	0.615	0.2302	0.8992	0.7698	0.8295	0.6924	0.3247	0.3082
Class 1	0.615	0.2302	0.7698	0.385	0.3747	0.615	0.4657	0.6924	0.3247	0.3082
Wt. Average	0.7414	0.3566	0.6434	0.2586	0.8031	0.7414	0.7629	0.6924	0.3247	0.3082

- Oversampling + cfs + cforest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1582    |         436         |
| Negative |     920     |         1079        |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7837	0.4602	0.5398	0.2163	0.8837	0.7837	0.8307	0.6618	0.2801	0.2706
Class 1	0.5398	0.2163	0.7837	0.4602	0.3588	0.5398	0.4311	0.6618	0.2801	0.2706
Wt. Average	0.739	0.4155	0.5845	0.261	0.7876	0.739	0.7575	0.6618	0.2801	0.2706

- Oversampling + cfs + fda

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1566    |         452         |
| Negative |     778     |         1221        |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7758	0.3894	0.6106	0.2242	0.8989	0.7758	0.8328	0.6932	0.3277	0.3124
Class 1	0.6106	0.2242	0.7758	0.3894	0.3791	0.6106	0.4678	0.6932	0.3277	0.3124
Wt. Average	0.7455	0.3591	0.6409	0.2545	0.8037	0.7455	0.766	0.6932	0.3277	0.3124

- Oversampling + cfs + random forest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1614    |         404         |
| Negative |     1239    |         761         |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7996	0.6195	0.3805	0.2004	0.852	0.7996	0.825	0.5901	0.1647	0.1628
Class 1	0.3805	0.2004	0.7996	0.6195	0.2986	0.3805	0.3346	0.5901	0.1647	0.1628
Wt. Average	0.7228	0.5427	0.4573	0.2772	0.7506	0.7228	0.7352	0.5901	0.1647	0.1628

- Oversampling + cfs + CART

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1486    |         532         |
| Negative |     893     |         1106        |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7361	0.4469	0.5531	0.2639	0.8802	0.7361	0.8017	0.6446	0.2404	0.2254
Class 1	0.5531	0.2639	0.7361	0.4469	0.3197	0.5531	0.4052	0.6446	0.2404	0.2254
Wt. Average	0.7026	0.4134	0.5866	0.2974	0.7775	0.7026	0.7291	0.6446	0.2404	0.2254

- Oversampling + Information Gain + AdaBoost

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1606    |         412         |
| Negative |     893     |         1106        |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7956	0.4469	0.5531	0.2044	0.8882	0.7956	0.8394	0.6744	0.3045	0.2955
Class 1	0.5531	0.2044	0.7956	0.4469	0.3776	0.5531	0.4488	0.6744	0.3045	0.2955
Wt. Average	0.7512	0.4025	0.5975	0.2488	0.7947	0.7512	0.7679	0.6744	0.3045	0.2955

- Oversampling + Information Gain + bagFDA

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1558    |         460         |
| Negative |     716     |         1283        |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7718	0.3584	0.6416	0.2282	0.9057	0.7718	0.8334	0.7067	0.3477	0.3292
Class 1	0.6416	0.2282	0.7718	0.3584	0.3867	0.6416	0.4826	0.7067	0.3477	0.3292
Wt. Average	0.748	0.3346	0.6654	0.252	0.8106	0.748	0.7692	0.7067	0.3477	0.3292

- Oversampling + Information Gain + cforest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive |     1740    |         278         |
| Negative |     1044    |         955         |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.8621	0.5221	0.4779	0.1379	0.8804	0.8621	0.8712	0.67	0.3286	0.3282
Class 1	0.4779	0.1379	0.8621	0.5221	0.4372	0.4779	0.4566	0.67	0.3286	0.3282
Wt. Average	0.7917	0.4517	0.5483	0.2083	0.7992	0.7917	0.7953	0.67	0.3286	0.3282

- Oversampling + Information Gain + fda

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     1538    |         480           |
| Negative |     672     |         1327          |
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7619	0.3363	0.6637	0.2381	0.91	0.7619	0.8294	0.7128	0.3541	0.3321
Class 1	0.6637	0.2381	0.7619	0.3363	0.3846	0.6637	0.487	0.7128	0.3541	0.3321
Wt. Average	0.7439	0.3183	0.6817	0.2561	0.8138	0.7439	0.7667	0.7128	0.3541	0.3321

- Oversampling + Information Gain + random forest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     1940    |         78            |
| Negative |     1424    |         575           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.9613	0.7124	0.2876	0.0387	0.8575	0.9613	0.9064	0.6245	0.3466	0.3148
Class 1	0.2876	0.0387	0.9613	0.7124	0.625	0.2876	0.3939	0.6245	0.3466	0.3148
Wt. Average	0.8379	0.589	0.411	0.1621	0.8149	0.8379	0.8125	0.6245	0.3466	0.3148

- Oversampling + Information Gain + CART

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     1468    |         550           |
| Negative |     911     |         1088          |
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7272	0.4558	0.5442	0.2728	0.8768	0.7272	0.795	0.6357	0.2246	0.2096
Class 1	0.5442	0.2728	0.7272	0.4558	0.309	0.5442	0.3942	0.6357	0.2246	0.2096
Wt. Average	0.6937	0.4223	0.5777	0.3063	0.7728	0.6937	0.7216	0.6357	0.2246	0.2096

- Undersampling + Boruta + AdaBoost

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     330     |         119         |
| Negative |     153     |         297         |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7351	0.3407	0.6593	0.2649	0.9059	0.7351	0.8116	0.6972	0.3227	0.2974
Class 1	0.6593	0.2649	0.7351	0.3407	0.3582	0.6593	0.4642	0.6972	0.3227	0.2974
Wt. Average	0.7212	0.3268	0.6732	0.2788	0.8056	0.7212	0.748	0.6972	0.3227	0.2974

- Undersampling + Boruta + bagFDA

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     324     |         125         |
| Negative |     151     |         299         |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7212	0.3363	0.6637	0.2788	0.9054	0.7212	0.8029	0.6925	0.3123	0.2848
Class 1	0.6637	0.2788	0.7212	0.3363	0.348	0.6637	0.4566	0.6925	0.3123	0.2848
Wt. Average	0.7107	0.3258	0.6742	0.2893	0.8033	0.7107	0.7395	0.6925	0.3123	0.2848

- Undersampling + Boruta + cforest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     320     |         129         |
| Negative |     161     |         289         |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7113	0.3584	0.6416	0.2887	0.8985	0.7113	0.794	0.6765	0.2856	0.2594
Class 1	0.6416	0.2887	0.7113	0.3584	0.3326	0.6416	0.4381	0.6765	0.2856	0.2594
Wt. Average	0.6985	0.3456	0.6544	0.3015	0.7949	0.6985	0.7288	0.6765	0.2856	0.2594

- Undersampling + Boruta + fda

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     318     |         131         |
| Negative |     149     |         301         |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7073	0.3319	0.6681	0.2927	0.9048	0.7073	0.794	0.6877	0.3023	0.2726
Class 1	0.6681	0.2927	0.7073	0.3319	0.3386	0.6681	0.4494	0.6877	0.3023	0.2726
Wt. Average	0.7001	0.3247	0.6753	0.2999	0.8011	0.7001	0.7309	0.6877	0.3023	0.2726

- Undersampling + Boruta + random forest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     327     |         122         |
| Negative |     179     |         271         |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7282	0.3982	0.6018	0.2718	0.8908	0.7282	0.8013	0.665	0.2709	0.2508
Class 1	0.6018	0.2718	0.7282	0.3982	0.3317	0.6018	0.4277	0.665	0.2709	0.2508
Wt. Average	0.7051	0.3751	0.6249	0.2949	0.7884	0.7051	0.7329	0.665	0.2709	0.2508

- Undersampling + Boruta + CART

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     304     |         145           |
| Negative |     159     |         291           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.6756	0.354	0.646	0.3244	0.8949	0.6756	0.7699	0.6608	0.2559	0.2259
Class 1	0.646	0.3244	0.6756	0.354	0.3087	0.646	0.4178	0.6608	0.2559	0.2259
Wt. Average	0.6702	0.3486	0.6514	0.3298	0.7875	0.6702	0.7054	0.6608	0.2559	0.2259

- Undersampling + cfs + AdaBoost

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     313     |         136           |
| Negative |     153     |         297           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.6964	0.3407	0.6593	0.3036	0.9012	0.6964	0.7857	0.6779	0.2852	0.2554
Class 1	0.6593	0.3036	0.6964	0.3407	0.3275	0.6593	0.4376	0.6779	0.2852	0.2554
Wt. Average	0.6896	0.3339	0.6661	0.3104	0.7961	0.6896	0.7219	0.6779	0.2852	0.2554

- Undersampling + cfs + bagFDA

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     315     |         134           |
| Negative |     155     |         295           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7014	0.3451	0.6549	0.2986	0.9006	0.7014	0.7886	0.6781	0.2864	0.2576
Class 1	0.6549	0.2986	0.7014	0.3451	0.3296	0.6549	0.4385	0.6781	0.2864	0.2576
Wt. Average	0.6929	0.3366	0.6634	0.3071	0.796	0.6929	0.7245	0.6781	0.2864	0.2576

- Undersampling + cfs + cforest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     323     |         126           |
| Negative |     167     |         283           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7192	0.3717	0.6283	0.2808	0.8962	0.7192	0.798	0.6738	0.2829	0.2591
Class 1	0.6283	0.2808	0.7192	0.3717	0.3341	0.6283	0.4362	0.6738	0.2829	0.2591
Wt. Average	0.7026	0.3551	0.6449	0.2974	0.7933	0.7026	0.7317	0.6738	0.2829	0.2591

- Undersampling + cfs + fda

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     320     |         129           |
| Negative |     163     |         287           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7123	0.3628	0.6372	0.2877	0.8975	0.7123	0.7942	0.6747	0.2831	0.2575
Class 1	0.6372	0.2877	0.7123	0.3628	0.3318	0.6372	0.4364	0.6747	0.2831	0.2575
Wt. Average	0.6985	0.349	0.651	0.3015	0.7939	0.6985	0.7287	0.6747	0.2831	0.2575

- Undersampling + cfs + random forest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     368     |          81           |
| Negative |     213     |         237           |
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.8194	0.4735	0.5265	0.1806	0.8853	0.8194	0.8511	0.673	0.3116	0.3065
Class 1	0.5265	0.1806	0.8194	0.4735	0.3953	0.5265	0.4516	0.673	0.3116	0.3065
Wt. Average	0.7658	0.4199	0.5801	0.2342	0.7956	0.7658	0.7779	0.673	0.3116	0.3065

- Undersampling + cfs + CART

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     318     |          131          |
| Negative |     189     |         261           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7083	0.4204	0.5796	0.2917	0.8826	0.7083	0.7859	0.644	0.2344	0.2147
Class 1	0.5796	0.2917	0.7083	0.4204	0.3082	0.5796	0.4024	0.644	0.2344	0.2147
Wt. Average	0.6847	0.3968	0.6032	0.3153	0.7774	0.6847	0.7157	0.644	0.2344	0.2147

- Undersampling + Information Gain + AdaBoost

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     317     |          132          |
| Negative |     151     |         299           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7054	0.3363	0.6637	0.2946	0.9034	0.7054	0.7922	0.6845	0.297	0.2676
Class 1	0.6637	0.2946	0.7054	0.3363	0.3356	0.6637	0.4458	0.6845	0.297	0.2676
Wt. Average	0.6978	0.3287	0.6713	0.3022	0.7994	0.6978	0.7288	0.6845	0.297	0.2676

- Undersampling + Information Gain + bagFDA

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     335     |         114           |
| Negative |     167     |         283           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.746	0.3717	0.6283	0.254	0.8995	0.746	0.8156	0.6872	0.3098	0.289
Class 1	0.6283	0.254	0.746	0.3717	0.3568	0.6283	0.4551	0.6872	0.3098	0.289
Wt. Average	0.7244	0.3501	0.6499	0.2756	0.8001	0.7244	0.7496	0.6872	0.3098	0.289

- Undersampling + Information Gain + cforest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     323     |         126           |
| Negative |     145     |         305           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7192	0.323	0.677	0.2808	0.9085	0.7192	0.8028	0.6981	0.3206	0.2913
Class 1	0.677	0.2808	0.7192	0.323	0.3509	0.677	0.4622	0.6981	0.3206	0.2913
Wt. Average	0.7115	0.3153	0.6847	0.2885	0.8064	0.7115	0.7404	0.6981	0.3206	0.2913

- Undersampling + Information Gain + fda

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     340     |         109           |
| Negative |     153     |         297           |
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7569	0.3407	0.6593	0.2431	0.9083	0.7569	0.8257	0.7081	0.3453	0.3231
Class 1	0.6593	0.2431	0.7569	0.3407	0.3782	0.6593	0.4807	0.7081	0.3453	0.3231
Wt. Average	0.739	0.3228	0.6772	0.261	0.8112	0.739	0.7625	0.7081	0.3453	0.3231

- Undersampling + Information Gain + random forest

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     330     |         119           |
| Negative |     157     |         293           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7351	0.3496	0.6504	0.2649	0.9037	0.7351	0.8107	0.6928	0.3158	0.2915
Class 1	0.6504	0.2649	0.7351	0.3496	0.3551	0.6504	0.4594	0.6928	0.3158	0.2915
Wt. Average	0.7196	0.3341	0.6659	0.2804	0.8032	0.7196	0.7464	0.6928	0.3158	0.2915

- Undersampling + Information Gain + CART

```
Confusion Matrix:
-------------------------
|          |  Predicted  |
|          |  Positive   |  Predicted Negative |
|----------|-------------|---------------------|
| Positive   |     317     |         132           |
| Negative |     173     |         277           |
-------------------------
```

	TPR	FPR	TNR	FNR	Precision	Recall	F-measure	ROC	MCC	Kappa
Class 0	0.7054	0.385	0.615	0.2946	0.891	0.7054	0.7874	0.6602	0.2593	0.2355
Class 1	0.615	0.2946	0.7054	0.385	0.3188	0.615	0.4199	0.6602	0.2593	0.2355
Wt. Average	0.6888	0.3684	0.6316	0.3112	0.7862	0.6888	0.7201	0.6602	0.2593	0.2355

The best 3 combinations according to high class 0 (Y) TPR AND Class 1 (N) TPR:

1. Oversampling + Information Gain + fda:

class Y TPR = 76.19% AND class N TPR = 66.37%

2. Oversampling + Boruta + fda

class Y TPR = 75.6% AND class N TPR = 67.26%

3. Undersampling + Information Gain + fda

class Y TPR = 75.69% AND class N TPR = 65.93%

NOTE: The above results were only recreated by running all the 36 combinations together. Running individual models will give us slightly different results because of multiple seed initializations in the for loop. So, these results cannot be recreated if an individual models are run separately. The best model in R code is run separately and has slightly different results than those mentioned above. So that model should be considered for external dataset.

We have considered the best model to be *Undersampling + Information Gain + fda.* This is because Undersampling has a low chance of overfitting, and both undersampling and Information Gain are more time efficient than Oversampling and Boruta. So, for a small tradeoff in accuracy, we chose it as the best combination of training.

The results of our experiment demonstrated significant variations in performance across different combinations. Some key findings include:

- Oversampling combined with Boruta and various classification algorithms improved classification performance, particularly in terms of TPR, Precision, and F-measure compared to other combinations.
- Random Forest exhibited strong performance across multiple combinations, achieving high TPR and Precision.
- AdaBoost and BagFDA also showed promising results, especially when paired with oversampling and Boruta feature selection.
- Feature selection methods of Boruta and Information Gain contributed to enhancing classification performance in most cases.

Discussion:

- The choice of balancing method, feature selection technique, and classification algorithm significantly impacts classification performance.
- Oversampling techniques tend to improve performance by mitigating class imbalance, but they may lead to overfitting in some cases.

- Boruta and Information Gain feature selection methods effectively identify relevant features and enhance classification accuracy.
- Random Forest emerges as a robust classifier, demonstrating consistent performance across various combinations.

Conclusion: In conclusion, our project experiment highlights the importance of selecting appropriate methodologies for handling imbalanced datasets and improving classification performance. Further exploration and fine-tuning of these methodologies could lead to more effective solutions for real-world classification tasks. Overall, this study provides valuable insights for practitioners in the field of machine learning and data mining.

Division of work: Most of the work was done colaberatively, but the major duty was:

Documentation + Preprocessing: Sarthak Mahapatra

Training new algorithms: Nahush Bhamre