Assessing Feature Selection Methods and Their Performance in High Dimensional Classification Problems

by

Mathara Arachchige Dona Surani Lakshima

A Thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

Master of Science

Department of Statistics

University of Manitoba

Winnipeg

Abstract

High dimensional classification problems have gained increasing attention in machine learning, and feature selection has become an essential step in executing machine learning algorithms. Identifying the smallest feature subset with the most informative features is the most crucial objective in feature selection. First, we propose an extended version of wrapper feature selection methods, which selects a further smaller feature subset yet with similar performance. Secondly, we examine four existing feature ordering techniques to find the most informative ordering mechanism. Using the results, we suggest a better method by combining a sequential feature selection technique with the sum of absolute values of principal component loadings to get the most informative subset of features. We further merge two different proposed approaches and compare the performance with the existing Recursive Feature Elimination (RFE) by simulating data for several practical scenarios with a different number of informative features, sample sizes, and different imbalance rates. We also use the Synthetic Minority Oversampling Technique (SMOTE) to analyze the behavior of the proposed approach. Our simulated results and application results show that the proposed methods outperform the original RFE by giving a reasonable increment or an insignificant reduction of F1-score on various data sets.

Keywords: Feature Selection, Wrapper Methods, Recursive Feature Elimination, Class Imbalance, SMOTE, Principal Component Loadings

Acknowledgment

Foremost I would like to express my sincere gratitude to my advisors Dr. Saman Muthukumarana and Dr. Mike Domaratzki, whose invaluable support, enthusiasm, guidance and immense knowledge towards my MSc. thesis from the beginning of my research work.

I would also like to thank Dr. Max Turgeon and Dr. Carson Kai-Sang Leung for being in the advisory committee and allocating their time to review and help me to complete this thesis.

Many thanks go to the staff, support staff, the faculty and the colleagues in the Department of Statistics, University of Manitoba. Also, I would like to thank the Department of Statistics, University of Manitoba for the financial support provided throughout my MSc. study.

Finally and most specially, I would like to thank my parents, my beloved husband Sudesh for their support, patience and tolerance during past years to make my thesis a success.

Dedication Page

This work is dedicated to my mother, father and my loving husband who have supported me with heart and soul.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

With the immense development of machine learning concepts and related topics, the attention towards feature selection has become one of the most crucial aspects of successful analysis because most real-world data sets suffer from many features. This problem is known as the curse of dimensionality (Bellman, 1957), and many sectors negatively experience this issue, including in the worlds of business, industry, and scientific research.

Selecting a few features is known as feature selection. Before fitting a model, we prefer the most informative features to be included in the data set while discarding the least informative features. Therefore, feature selection should remove non-informative features from the model (Kuhn, 2013). Feature selection provides several significant advantages. With feature selection, dimensionality reduction can decrease the size of the data without harming the overall performance of the analytical algorithm (Nisbet, 2012). The decrease of computational time while increasing the algorithm's predictive power and interpretability are notable gains (Miche et al., 2007a; Samb et al., 2012). Then again, a model with fewer features may be less costly, especially if there is a

cost of measuring the features (Ali et al., 2020). Statistically, it is more convenient and attractive to estimate fewer parameters, and it will also reduce the negative impact of non-informative features. Further, it becomes increasingly challenging to reveal patterns in data with many features (Guo et al., 2002).

1.1 Motivation

Two main objectives of feature selection are

1. Reduce the number of features
2. Identify the most informative feature subset.

Therefore, the motivation of the research is to address both of these objectives and finally introduce a method that selects the most informative, minimal number of features with similar performance.

Although we can already reduce the number of features using wrapper feature selection methods such as Recursive Feature Elimination (RFE) according to a given selection scoring criteria, there is room to improve it further. We observed that the number of features of the chosen subset by RFE might not be the expected one if the desire is having a smaller number of features. In particular, there are other selections of a lower number of features with negligibly lower model accuracy. Therefore, we consider the challenge of finding an optimal threshold to identify this minuscule difference.

Fig. 1.1 shows an example of the effect of feature selection using RFE, a common algorithm. It is a plot of the cross validation F1-scores of different-sized feature subsets, and we can clearly see that RFE (blue point) has selected

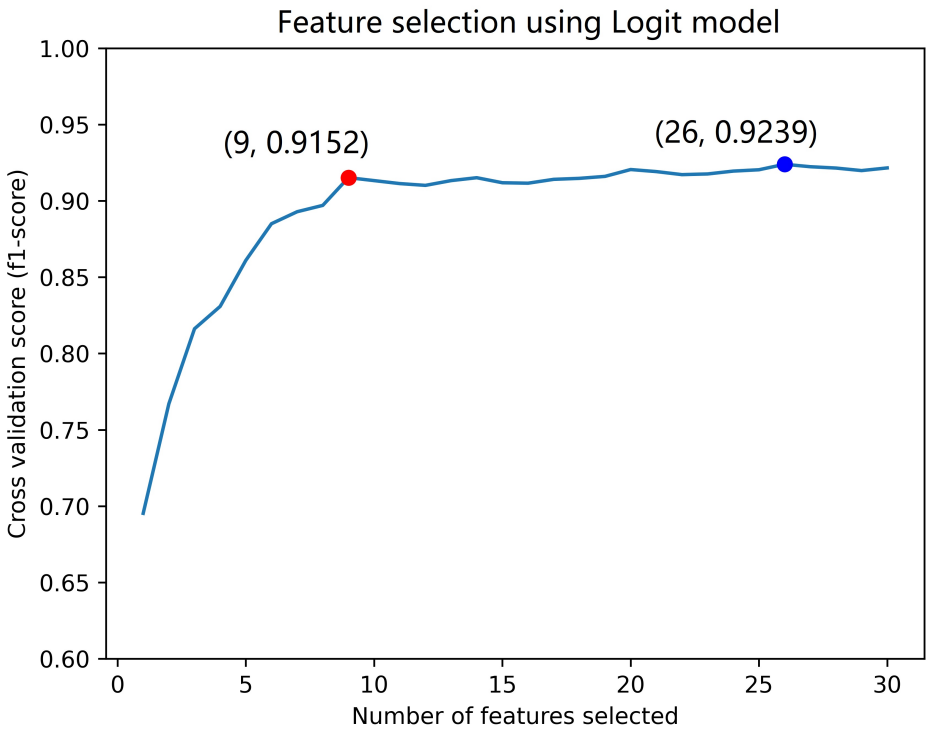

Fig. 1.1. The blue point indicates the RFE feature selection with number of selected features and the F1-score while the red point explains the same for the proposed method.

26 features, but the F1-score does not appear to be much improved after around nine features. Meanwhile, the method proposed in this thesis (red point) suggests nine features as the smaller number of features with similar performance.

There are several feature selection techniques in the literature, yet they behave differently with varying data sets. In classification problems, when the classes are imbalance, and the data set suffers from many non-informative features, most feature selection techniques may not accurately select all infor-

mative features. Hence, the resulting output may not be the one expected. Therefore, identifying the most suitable feature ordering and feature selection technique is a significant concern requiring a solidified solution. Hence, we will investigate the impact of the number of informative features and the sample size of the imbalanced data set on feature selection techniques.

Our primary focus is analyzing the behavior of wrapper methods towards classification accuracy and suggesting a better extension for selecting a smaller number of features with similar performance with the previous method. Hence, we introduce an algorithm with a threshold to achieve this objective. Besides choosing the minimal number of features, we suggest the appropriate feature subset by considering feature importance. We also compare different feature ordering techniques to identify the best and use a sequential feature selection method to pick the most informative features from the data set. Finally, combining two proposed methods we return the most informative feature subset with minimal number of features with better performance.

To cover most practical scenarios, we first synthetically simulated data using "make classification" library in Python scikit-learn.datasets (Pedregosa et al., 2011a) and compare the performance of existing and the proposed method. Later, we apply both methods in real-world data sets to derive further conclusions. We also use a re-sampling technique, Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002) to determine the optimal performance of the machine learning model.

1.2 Thesis Overview

We conduct a literature review to emphasize the background information about the research, and the detailed discussion is in Chapter 2. In Chapter 3 we suggest a method to select a minimal number of features with similar performance. The introduction to the new method, related work, and synthetic and application results will also be presented in the same Chapter. Then Chapter 4 explores another aspect of the feature selection by introducing the principal component loading feature selection method. Simulation and application results of the combination of two proposed approaches will also be explained and validated using existing methods in Chapter 5. Finally, we conclude the thesis with a discussion on our results and solutions to the problem of interest in Chapter 6.

Chapter 2

Background

2.1 Chapter Overview

In this chapter, we discuss the importance of feature selection (section 2.2) in high dimensional situations and the role of class imbalance data (section 2.3). To assess the success of the proposed and existing strategies, we use several classification models and several model metrics to compare the results. These classifiers are discussed in section 2.4 and classification-based model importance is explained in section 2.5. Trained models are then appropriately evaluated in the thesis using several model metrics discussed in section 2.6. Furthermore, one of the proposed methods is based on Principal Component Analysis (PCA); thus, it is discussed in section 2.7 in detail.

2.2 Feature Selection

Feature selection determines which features should be included in a model, and it is becoming one of the most critical questions with the curse of dimensionality; data are becoming increasingly high-dimensional. Basically, we prefer the most informative features to be selected and want to discard the least informative features in feature selection. The need for feature selection arises with some significant advantages (Miche et al., 2007b).

- Train the machine learning algorithm faster.
- Reduce the complexity of a model.
- Decrease the computational time.
- Easier to interpret the model.
- Build a sensible model with better prediction power.
- Reduce over-fitting by selecting the right set of features.

There are mainly three categories of feature selection methods introduced in the literature: filter, wrapper, and embedded methods (Stańczyk, 2015; Lal et al., 2006).

2.2.1 Filter methods

Filter methods measure the relevance of features by their correlation with the dependent variable; hence, only features with meaningful relationships would be included in a classification model. Filter methods use statistical methods such as Pearson's Correlation, Analysis of Variance (ANOVA), Linear discriminant analysis (LDA), and Chi-Squared statistics to evaluate a subset of features.

2.2.2 Wrapper methods

Wrapper methods measure the usefulness of a subset of features by actually training a model on it (Saeys et al., 2007). Some wrapper methods perform this evaluation with different randomly selected subsets, using a cross-validation (CV) method, which is described in more detail in section 2.6.1. Cross-validation divides the data set into several subsets for each group of features and evaluates a model trained on all but one subset (Nisbet, 2009). The basic process of wrapper methods is shown in Fig. 2.1. Forward Feature Selection, Backward Feature Elimination (Weisberg, 2005), and Recursive Feature Elimination (RFE) (Guyon et al., 2002) are typical examples of commonly used wrapper methods.

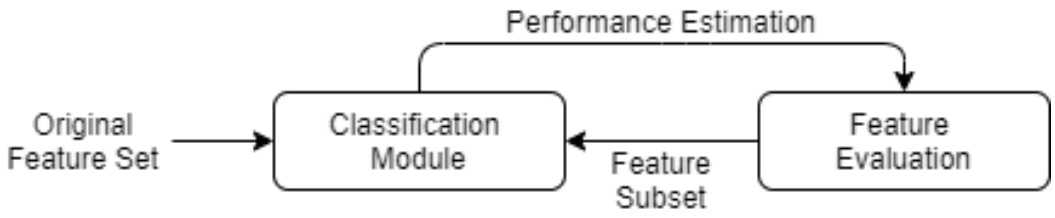

Fig. 2.1. Process of wrapper feature selection.

2.2.3 Embedded methods

The third category, embedded methods, is quite similar to wrapper methods. Although the embedded methods also optimize the objective function or performance of a learning algorithm or model, they also use an intrinsic model-building metric during learning. L1 (LASSO) regularization (Tibshirani, 1996) and Elastic Net (Zou and Hastie, 2005) are commonly known embedded methods.

Table 2.1: Differences between Filter, Wrapper and Embedded methods

Filter Method	**Wrapper Method**	**Embedded Method**
A generic set of methods that use statistical methods for evaluation of a subset of features.	Evaluates on a specific machine-learning algorithm to find optimal features and use cross validation.	Embeds features during model building process. Feature selection is done by observing each iteration of model training phase.
Much faster compared to wrapper methods in terms of time complexity.	Computationally very expensive for a data set with many features.	Time complexity is in between filter and wrapper methods.
Independent of classifier.	Interacts with classifier.	Interacts with classifier.
Eg: Chi-square, Euclidean distance, Correlation based feature selection, Markov blanket filter, ANOVA Fast correlation based feature selection	Eg: Sequential forward selection, Sequential backward selection, Randomized hill climbing, Recursive Feature Elimination	Eg: LASSO regularization, Elastic Net

A comparison between three feature selection categories is presented in Table 2.1 (Kumari and Swarnkar, 2011). In this research, we mainly consider the wrapper methods (Kohavi and John, 1997), which iteratively examine different subsets to obtain improved accuracy on fewer features. RFE (Guyon et al., 2002) is one such commonly used technique. In standard RFE, a feature is eliminated if it is the least important to predicting, and features are ranked according to the model's strength by considering the performance scoring method. Basically, RFE obtains the global value of the cross-validation score of each subset by maximizing the F1-score over k fold cross-validation and returns the subset with the best value.

In 2005, Hua at al. (Hua et al., 2005) tried to find the optimal number of features as a function of sample size for various classification rules and derived two main conclusions. First, optimal-feature-size behavior relative to the sample size depends strongly on the classifier and the feature-label

distribution. Second, the number of features can significantly influence the performance of a classifier.

2.3 Class Imbalance

Prior research compares the impact of class re-balancing techniques on the performance of binary prediction models for a different choice of data sets, classification techniques, and performance measures.

We focus on binary classification problems where there are only two possible outcomes. For example, consider trying to predict customer churn, where the two outcomes are churners and non-churners. The class imbalance issue occurs when the number of instances in the small (minority) class is significantly smaller than the number of instances in the large (majority) class. It produces a significant negative influence on standard classification learning algorithms. Reasonably, the minority class is more relevant and more important in practical situations; therefore, it requires an intense urgency to be identified (Sun et al., 2009). Indeed, the problem of learning on imbalanced data sets is considered to be one of the ten challenging issues in data mining research, as listed in (Yang and Wu, 2006).

However, in such cases, standard classifiers tend to be overwhelmed by the majority classes and ignore the minority classes (Wu and Chang, 2003). As mentioned earlier this problem is widespread in many applications, including: anomaly detection (Tavallaee et al., 2010), fraud/intrusion detection (Yang et al., 2009), medical diagnosis/monitoring (Mazurowski et al., 2008), and churn prediction (Zhu et al., 2018). However, studies on class imbalance classification have gained more emphasis only in recent years (Kotsiantis et al., 2005),

and many solutions have been proposed to eliminate the problem of learning from imbalanced data sets. The most popular proposed solutions can be categorized into two levels: data-level and algorithm-level. Data-level solutions (Ganganwar, 2012) apply resampling as a preprocessing step to reduce the negative effect caused by class imbalance. These methods include random and focused under/oversampling methods and synthetic data generation methods like SMOTE (Synthetic Minority Oversampling TEchnique) (Chawla et al., 2002). Algorithm-level (Chawla et al., 2004) solutions aim to develop new algorithms or modify existing ones to bias learning towards the minority class. Ensemble learning methods are also another concept discussed in recent papers. Generally, in these methods, the majority class data set is separated into multiple sub-datasets. Each of these sub-datasets has a similar number of examples as the minority class data set.

However, each solution has pros and cons, without explicit agreement on what makes for the best solution to address each situation. Tantithamthavorn et al compared the impact of different class re-balancing techniques on commonly-used performance measures and interpreting the defect prediction models (Tantithamthavorn et al., 2018). In conclusion, they suggest that AUC should be used as a standard measure for comparing defect prediction models.

Our research will mainly use SMOTE, a data-level solution, as a re-balancing technique to achieve higher accuracy in applications. We do not consider algorithm-level solutions in this thesis.

2.3.1 Synthetic Minority Oversampling TEchnique (SMOTE)

As mentioned in the section 2.3, there are several resampling techniques in literature, and some of the common approaches would be under-sampling and over-sampling. In most cases, over-sampling is preferred over under-sampling since by removing data, we might lose some informative instances. Still, random oversampling leads to over-fitting, which is another issue. Hence we choose SMOTE as the resampling technique, which allows us to generate synthetic samples for the minority class.

SMOTE works by selecting minority instances in the feature space, drawing a line between them, and drawing a new sample at a point along that line. In particular, first, it chooses a random point from the minority class. Then it finds k nearest neighbors for that point. A random neighbor is chosen, and a synthetic sample is created at a randomly selected point between the two points in the feature space.

Fig. 2.2 shows a hypothetical imbalanced data set with two features and the response consists of two classes. Minority class (1) instances appear in orange, whereas majority class (0) instances are represented in purple. Fig. 2.3 shows how SMOTE creates new minority class instances by identifying nearest neighbors. One possible balanced data set with synthetic data for the data displayed in Fig. 2.2 is shown in Fig. 2.4.

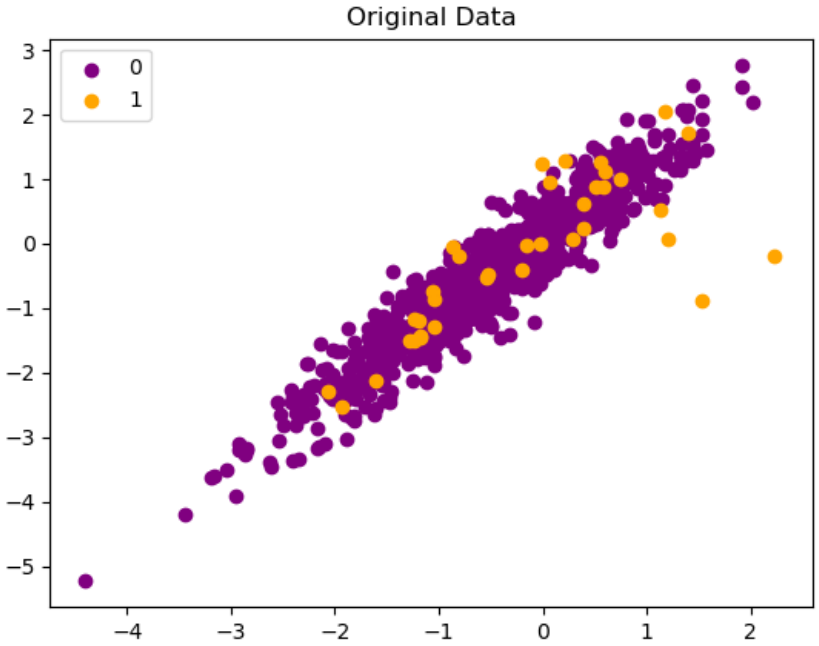

Fig. 2.2. Scatter plot for originally imbalanced data.

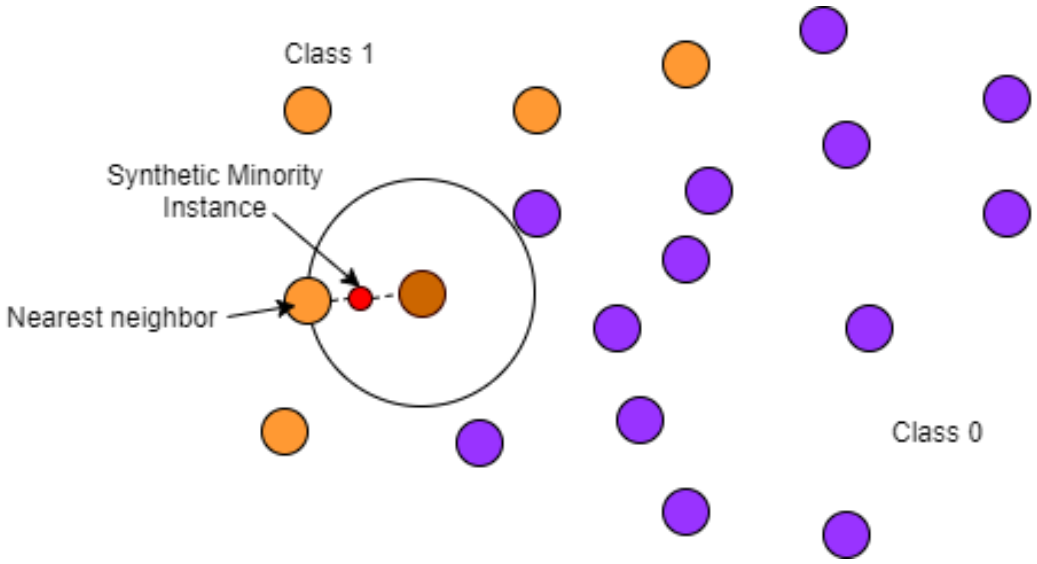

Fig. 2.3. Generating new instances between the nearest neighbor and the minority instance.

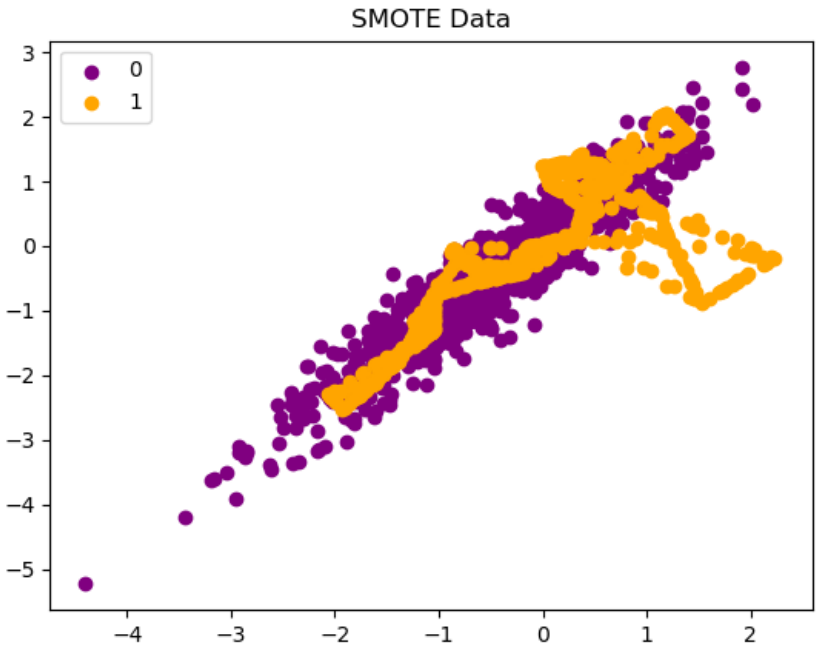

Fig. 2.4. Scatter plot for SMOTE data.

2.4 Machine Learning Classification Models

This section presents the five supervised learning binary classification models used to evaluate the performance changes over different factors, such as the number of informative features and imbalance rate. We use Logistic Regression (LOGIT) (Weisberg, 2005), Linear Support Vector Machine (SVM-Linear) (Xia and Jin, 2008), Decision Tree (Guo et al., 2002; Breiman et al., 1984), Random Forest (RFC) (Breiman, 2001), and Light Gradient Boosting (LGBM) (Friedman, 2001) to compare the impact of each method used.

2.4.1 Logistic regression

Logistic regression (a binomial logit model) is one of the most common methods to make predictions for binary classification problems (McCullagh and Nelder, 1989; Hastie et al., 2009). In the literature, this model is also known as logit regression, maximum-entropy classification (MaxEnt), or the log-linear classifier. It is a statistical model whose basic form uses a logistic function to model a binary dependent variable and find the probability of two binary outcomes.

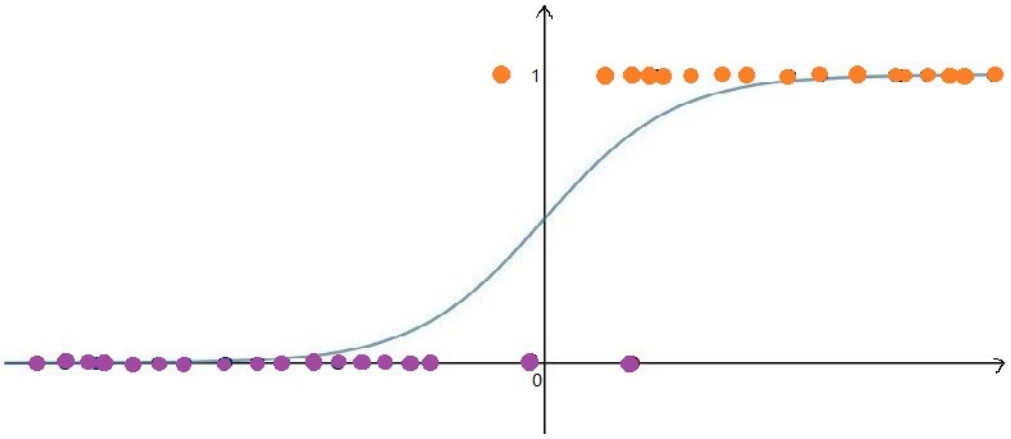

Fig. 2.5. Logistic regression curve.

$$h_\theta(x) = \frac{1}{1 + e^{-\beta x}} \tag{2.1}$$

As shown in Fig. 2.5, the logistic regression cost function has a sigmoid curve, which can be written as in equation 2.1. The curve represents the probability of a certain class or event existing. Values of y can be denoted by "1" (minor) and "0" (major) and are shown in orange and purple respectively. In this simple example, x is the only feature, and y is the predicted probability of the outcome. In the case of applying Logistic Regression to multiple classes

(for example, two classes), we can consider the predicted value of $y \in 0, 1$, based on the input x, using a threshold classifier: if $h_\theta(x) \geq 0.5$, then $y = 1$, else $y = 0$.

Logistic regression is a simple method to implement, interpret, and very efficient to train. But the major limitation of Logistic Regression is the assumption of linearity between the dependent and independent variables on the logit scale.

2.4.2 Support vector machine

A supervised machine learning algorithm, Support Vector Machine (SVM) is used as a binary classification technique. It is a general learning algorithm based on statistical learning theory, which can solve the nonlinearity, high dimension, and local minimization problems, which are insoluble for traditional methods (Xia and Jin, 2008).

In this algorithm, we plot each data point as a point in n-dimensional space, where n is the number of features. Then, by finding the hyperplane that best differentiates the two classes, we perform classification. To find the optimal hyperplane that gives the best margin by maximizing the distance, SVM uses kernels, and for this research, we consider using linear kernel version of SVM as shown in Fig. 2.6. The new instances are classified according to what side of the hyperplane they are located on.

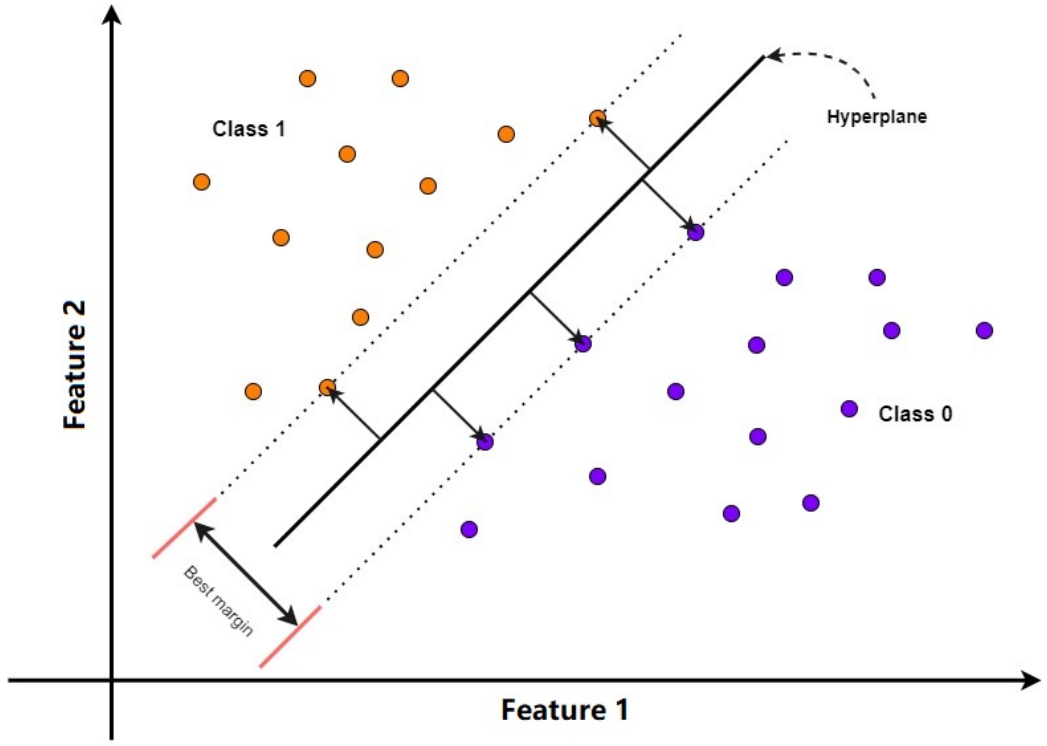

Fig. 2.6. Best-margin hyperplane and margins for an SVM trained with samples from two classes with two features.

2.4.3 Decision trees

Decision trees are a machine learning method that uses a series of decisions about individual features to predict an instance. During decision-making using a decision tree, multiple features participate, and it is necessary to consider the importance and relevance of each feature. Hence, having the most beneficial feature at the root node, the tree traverses downwards by splitting into branches and internal nodes. The most basic example of decision tree algorithms is the C4.5 (Quinlan, 1993) which uses several splitting measures like Entropy, Information Gain, and Gini Index to recursively split the available features into

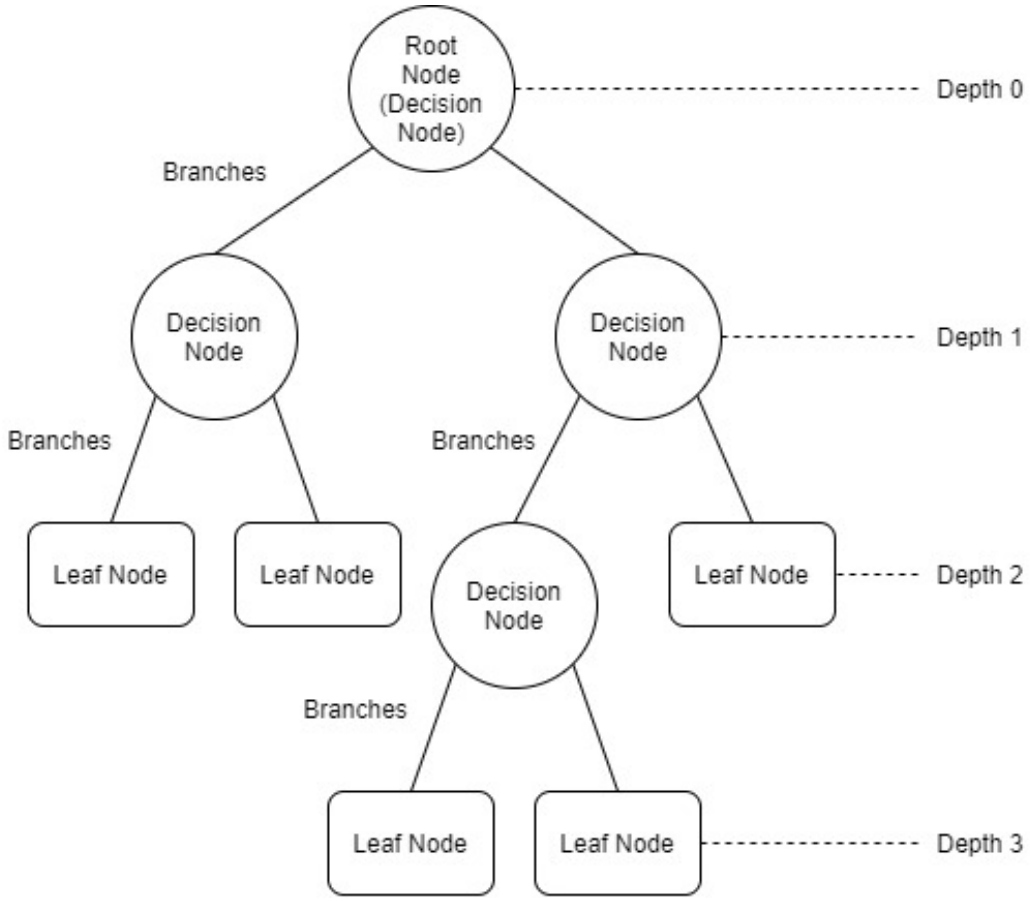

Fig. 2.7. Traditional decision tree layout with depth of 3.

decisions that will decrease the entropy across all values associated with that feature. This leads to a decrease in impurity and uncertainty and yields better classification at each node. In this thesis, we use Gini Index as the splitting measure.

Gini Index, also known as Gini impurity, calculates the probability of incorrectly classifying a specific feature when selected randomly.

$$Entropy = H(D) = -\sum_{i=1}^{k} p_i log(p_i) \tag{2.2}$$

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2 \tag{2.3}$$

where,

D : Data set

k : Number of classes (Here, $k = 2$)

p_i : Probability of a point belonging to class i

Fig. 2.7 shows an example visualization for a decision tree with a depth of three.

2.4.4 Random forest classifier

Random forest is also an easy-to-use supervised learning machine learning algorithm. The "forest" it builds is a collection of some decision trees on various sub-samples of the data set and merges them together to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size, but the samples are drawn with replacement.

The random forest classifier has nearly the same hyper-parameters as a decision tree. In addition, it adds additional randomness to the model when growing the trees. Instead of searching for the most important feature, a random forest classifier searches for the most important feature among a random subset of features while splitting a node, which generally results in a better model.

Fig. 2.8 shows how a random forest would look like with two trees. Here, each tree gives an independent prediction, and then the majority prediction is the final prediction for the ensemble.

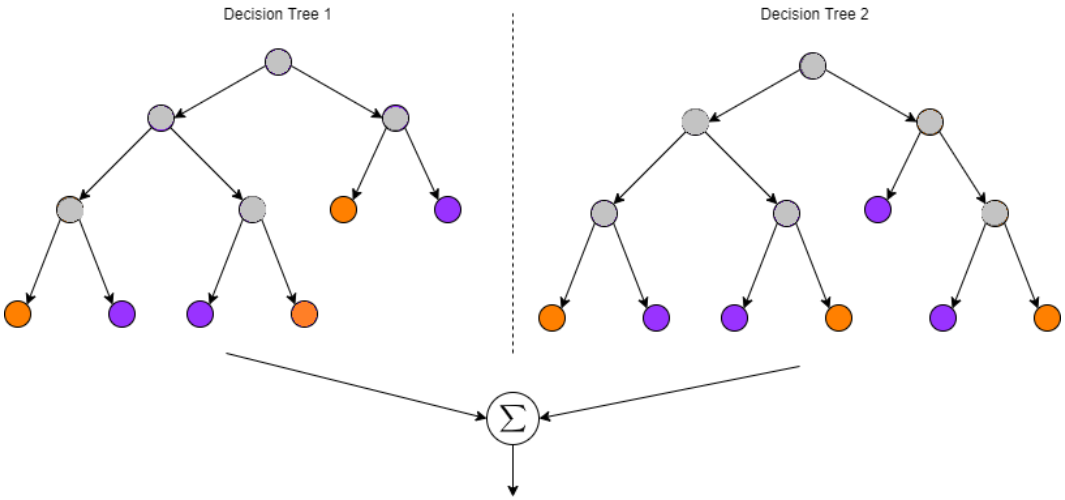

Fig. 2.8. Random forest classifier with two decision trees.

2.4.5 LightGBM classifier

Light Gradient Boosting Machine (Lgbm) is a gradient boosting algorithm that is also a tree-based learning algorithm. Light GBM grows trees vertically while other algorithms grow trees horizontally, meaning that Light GBM grows tree leaf-wise while other algorithms grow level-wise. It will choose the leaf with max delta loss to grow. In particular, leaf-wise chooses splits based on their contribution to the global loss and not just the loss along a particular branch. Hence, when growing the same leaf, the leaf-wise algorithm can reduce more loss than a level-wise algorithm. The visualization difference between the two methods is shown in Fig. 2.9.

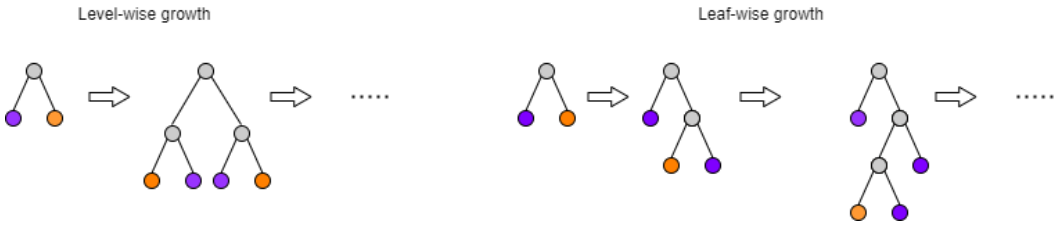

Fig. 2.9. Level-wise growth versus Leaf-wise growth of a decision tree.

2.5 Classification Model-Based Feature Importance

Classification model-based feature importance is a general technique of finding each feature importance using classification methods. It assigns scores to input features to a predictive model that indicates the relative importance of each feature when making a prediction. These feature importance scores can be calculated for regression problems as well as for classification problems. They are also useful in different situations in a predictive modeling problem, such as understanding the data and the model and reducing the number of input features for feature selection.

More specifically, we look at two main types of more advanced feature importance; they are feature importance from model coefficients and feature importance from decision trees.

2.5.1 Feature importance from model coefficients

Linear machine learning algorithms fit a model where the prediction is the weighted sum of the input values and find a set of coefficients to use in the

weighted sum to make a prediction (Tsuruoka et al., 2009). These coefficients can be used directly as a crude type of feature importance score. Examples include logistic regression and support vector machine with linear kernel.

Suppose the training sample is $(x_1, y_1), \ldots, (x_n, y_n)$ where $x_i \in \mathbf{R}^p$ and $y_i \in \{0, 1\}$. In this method, the goal is to learn a linear scoring function $f(x) = w^T x + b$ with model parameters $w \in \mathbf{R}^p$ and intercept $b \in \mathbf{R}$. To find the model parameters, we minimize the regularized training error given by:

$$E(w, b) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)) + \alpha R(w) \tag{2.4}$$

where, L is the loss function that measure the model misfit and R is a regularization term that penalizes model complexity. $\alpha > 0$ is a non-negative hyper-parameter that controls the regularization strength.

The loss function, L for the LOGIT classifier is,

$$L(y_i, f(x_i)) = log(1 + e^{-y_i f(x_i)}) \tag{2.5}$$

The loss function, L for the SVM-Linear classifier is,

$$L(y_i, f(x_i)) = max(0, 1 - y_i f(x_i)) \tag{2.6}$$

The p dimensional vector of model parameters w can be used to get feature importance. Large positive values of w_j signify higher importance of the j^{th} feature in the prediction of positive class, and large negative values signify higher importance in the prediction of negative class. Hence we consider taking the absolute values of the weights to have to the scale importance of each feature. In this research, we used `coef_` function given by Python scikit-learn (Pedregosa et al., 2011a) for each relevant model.

2.5.2 Feature importance from decision trees

We used the classification and regression trees (CART) (Ryzin, 1986) decision tree algorithm, which offers importance scores based on minimizing the criterion used to select split points, like Gini or entropy (see section 2.4.3).

At each node of a decision tree, the feature to be used for splitting the data set is decided based on information gain (IG) or the more computationally cheap Gini impurity reduction. The feature that maximizes IG (or reduction in Gini impurity) is selected as the splitting feature (f_i). Data is then divided amongst its children according to the value of the splitting feature. Information Gain due to a feature summed across all the levels of the decision tree determines its feature importance. This can also be seen from the fact that splitting is done on the feature that maximizes information gain at every node.

Hence, the information gain is defined as,

$$IG(D, f_j) = H(D) - \sum_{m=1}^{M} \frac{|D_m|}{|D|} H(D_m). \tag{2.7}$$

where,

M : Number of data sets obtained after splitting

D_m : m_{th}Data set obtained after splitting

We can use this same approach for collections of decision trees, such as the random forest and stochastic gradient boosting algorithms. Random forests comprise multiple decision trees; thus, the feature importance of feature j is the normalized sum of IG brought about by feature j across all trees.

In this research, we used `feature_importance_` function given by Python scikit-learn (Pedregosa et al., 2011a) for each relevant model which is used

Gini impurity reduction (explained in Section 2.4.3) as the criteria function to measure the quality of a split.

2.6 Performance Evaluation Matrices

To evaluate the best performing machine learning method with suggested and existing methods, we used precision, recall, and mainly the F1-score, which is derived using the model confusion matrices in Table 2.2.

Table 2.2: Confusion matrix for a binary classification problem

		Predicted	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

The F1-score is also called F-score or F-measure and is a measure of test accuracy in binary classification. In most real-life classification problems, imbalanced class distribution exists, and thus the F1-score is a suitable metric to evaluate the model (Provost and Fawcett, 2001). It is the harmonic mean of the precision and recall (Powers, 2008) and can be calculated as below.

$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{Recall} = \frac{TP}{TP + FN}$$
$$\text{F1-score} = 2\left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}\right)$$

The F1-score can be valuable in calculating the overall accuracy of a model; however, it is noticed that this calculation method does not take the true negative rate into account.

2.6.1 Cross-Validation (CV) technique

A number of model validation techniques have been developed over time, such as Re-substitution validation, Hold-out validation, k-fold cross-validation, Leave-one-out cross-validation, and Repeated k-fold cross-validation. Cross-validation (Refaeilzadeh et al., 2009) is a model validation technique, which is mainly used in practice to estimate how accurately a predictive model will perform. It divides the data set into several subsets for each group of features and evaluates a model trained on all but one subset.

K-fold cross-validation is a way to improve the basic idea. First, the data set is divided into k subsets, and each time, one of those k subsets is used as the test set, and the rest of the data is used as the training set to create the hypothesized model (Mitchell, 1997). Then the average error/accuracy (F1-score) across all k trials is computed.

One advantage of this method is that it matters less how the data gets divided. Each data point gets to be in a test set exactly once, and is in the training set $k - 1$ times. The variance of the resulting estimate is reduced when k is increased. However, this method is computationally expensive as the training algorithm has to be rerun k times, which means it takes k times as much computation to do the evaluation. Hence, k in the range 5-10 is usually sufficient.

2.7 Principal Component Analysis

Principal Component Analysis (Hotelling, 1933), or PCA, is a feature extraction method that is often used to reduce the dimensionality of large data sets by transforming a large set of features into a smaller set of variables that still contains most of the information in the large set. PCA reduces the number of features in the data set while preserving as large variation of the data as possible. The first principal component can equivalently be defined as a direction that maximizes the variance of the projected data (as in Fig. 2.10). The i^{th} principal component can be taken as a direction orthogonal to the first $i-1$ principal components that maximizes the variance of the projected data.

For a data set $\mathbf{X}$ of dimension n by p, PCA attempts to find linear orthogonal combinations of the columns of $\mathbf{X}$ with maximum variance such that $\sum_i A_i X_i = \mathbf{XA}$. After implementing PCA on the data set, the original features will turn into principal components, linear combinations of the original features. Hence, principal components are not as readable and interpretable as original features.

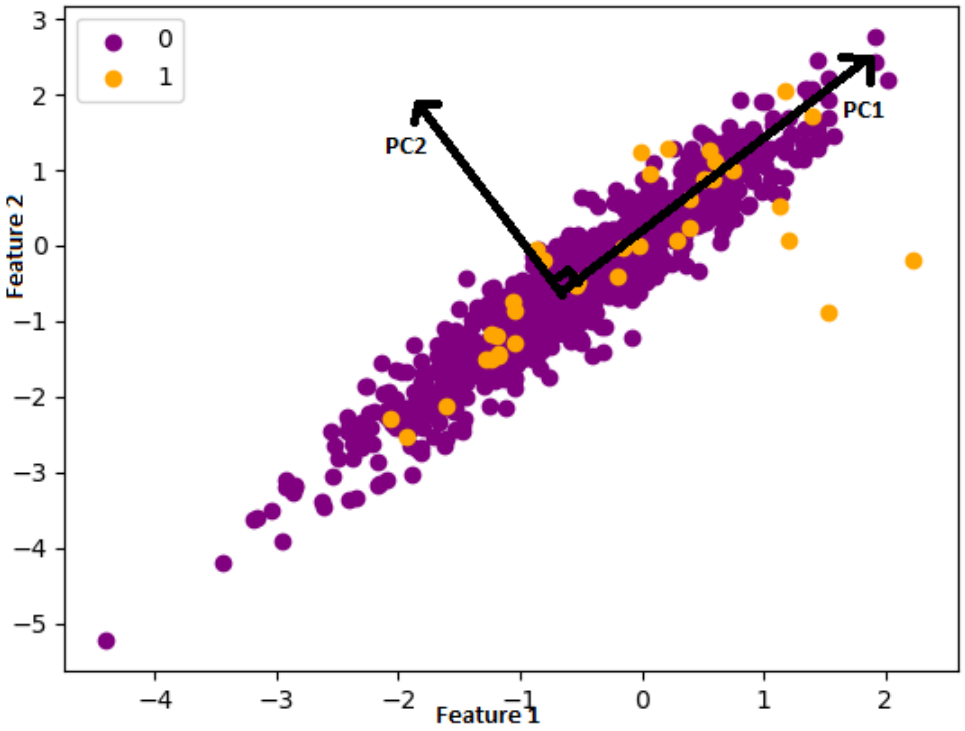

Fig. 2.10. First two Principal Components selections in two feature space.

Chapter 3

Selecting Features with Similar Performance

3.1 Chapter Overview

Most of the wrapper feature selection methods compare the score values of several feature subsets and select the one which gives the maximum score. There may be other selections of a lower number of features with a lower-scoring value, yet the difference is negligible. This Chapter proposes and applies an extended version of wrapper methods, which selects a smaller feature subset with similar performance to the original feature selection method under a pre-defined threshold. In section 3.4, we further validate the suggested extended version of the existing Recursive Feature Elimination (RFE) results by simulating data for several practical scenarios with a different number of features and different imbalance rates on several classification methods. The application results are presented in section 3.5.

3.2 Introduction

In this Chapter, we mainly consider the wrapper methods (Kohavi and John, 1997), which examine different subsets to achieve the objective of improved accuracy on fewer features. RFE (Guyon et al., 2002) is one such commonly used technique. It obtains the global value of the cross-validation score of each subset by maximizing the F1-score over k fold cross-validation and returns the subset with the best value (see Sect. 3.3).

To evaluate our new proposed extension, we synthetically simulate samples, where the sample size is 1000. The number of classes is two (binary classification), and there is only one cluster per class. Several numbers of features were considered to compare different situations. Since different classification models perform uniquely in different data sets, our goal is to introduce a general tool that performs with multiple models. To ensure this, we train different binary classification models in data sets with a different number of features and imbalance rates. Initially, five different binary classification models were trained with RFE. They are Logistic Regression (LOGIT) (McCullagh and Nelder, 1989; Hastie et al., 2009), Linear Support Vector Machine (SVM) (Cortes and Vapnik, 1995; Xia and Jin, 2008), Decision Tree (Guo et al., 2002; Breiman et al., 1984), Random Forest (RFC) (Breiman, 2001), and Light Gradient Boosting (LGBM) (Friedman, 2001).

3.3 Methods and Experimental Design

3.3.1 Recursive Feature Elimination (RFE)

RFE can be fitted on any classification model with an inherent quantification of the importance of a feature. It removes the weakest features by a step count, where the step is the number of features removed at each iteration. This process repeats until the stipulated number of features is reached. Features are ranked according to the importance identified by the model. Then, to find the optimal number of features, cross-validation is used in each iteration and selects the subset giving the best scoring value as the desired feature subset.

inputs:

Training samples: $\mathbf{X_0} = [X_1, X_2, \ldots, X_l]^T$

Class labels: $\mathbf{y} = [y_1, y_2, \ldots, y_l]^T$

outputs:

Feature ranked list: $\mathbf{r} = [r_1, r_2, \ldots, r_n]$

Grid scores: $\mathbf{g} = [g_1, g_2, \ldots, g_m]$

Number of selected features by RFE: n_{rfe}

Here, n is the number of features in the data set. Grid scores are the CV scores (e.g., F1-scores) such that g_i corresponds to the average CV score of the i^{th} feature subset with i remaining features where m is the total number of feature subsets. In our experiments, we are eliminating features by step of 1, and the minimum features to select is 1, hence $m = n$ (Pedregosa et al., 2011a).

3.3.2 Related work

Various approaches and extensions in the literature have been suggested to the existing feature selection mechanisms. Samb et al introduced an RFE-SVM-based feature selection approach by reusing previously removed features in RFE (Samb et al., 2012). They have used two local search tools, Bit-Flip (BF) and Attribute-Flip (AF), to improve the quality of the RFE. Nevertheless, this approach is specific for the SVM classification, where the method suggested in this thesis can be applied with any classification method, which facilitates a feature ranking criterion with feature importance.

An enhanced recursive feature elimination has been introduced by Chen and Jeong which is also an algorithm based on RFE and SVM (Chen and Jeong, 2007). It also assesses a weak feature removed by the standard RFE based on the classification performance before and after removing that feature and reconsidering it in the feature subset.

There are other proposed methods, which used thresholds to identifying the feature subset. A Receiver Operating Characteristic (ROC) curve based feature selection metric for small samples and imbalanced data (FAST) is recommended by Chen and Wasikowski (Chen and Wasikowski, 2008). This method is based on the area under a ROC curve by discretizing the distribution. An extension of the FAST method, but another threshold-based feature selection (TBFS) technique is discussed by Wang et al (Wang et al., 2010), where they produce 11 distinct versions of TBFS based on 11 different classifier performance metrics. These metrics look to modify the selection process to improve the selected subset rather than finding a smaller, non-optimal subset.

3.3.3 Suggested method

In this Chapter, we propose a new algorithm based on RFE. The suggested method is an extension of the RFE method, and the results that come out of the RFE algorithm are fed into the new algorithm to get the desired output. The main difference between the new method and the original RFE is that the original RFE chooses the feature subset giving the best scoring value in cross-validation. In contrast, the suggested method identifies a feature subset under an applicable threshold to obtain a smaller feature subset with similar performance and minimal loss. We compare RFE and the extended method on various synthetic data sets and show that the suggested method reduces the number of features with a bearable scoring value reduction. The algorithm for the new method is described below.

inputs:

Grid scores: $\mathbf{g} = [g_1, g_2, \ldots, g_m]$

Number of selected features by RFE: n_{rfe}

Total number of features: n

Feature importance scores (obtained from the classifier): $\mathbf{i} = [i_1, i_2, \ldots, i_{n_{rfe}}]$

Maximum tolerable F1-score reduction: T (User-defined)

procedure:

Step 1: As in Fig. 3.1, consider all the local maximum grid scores (g_j) corresponding to the number of subsets of features selected by RFE which is less than the optimal number of features selected (n_{rfe}) where,

$$g_j > max(g_{j-1}, g_{j+1}), \qquad j < n_{rfe}$$

Step 2: Connect each point with the maximum point and compute each line's gradient values (i.e., the tangent value of the cone).

Step 3: Compare the gradient values with a threshold value.

$$\text{gradient} = \frac{(\Delta y)_j}{(\Delta x)_j} < \text{Threshold}$$

The threshold (t) can be interpreted as the tolerable reduction of the F1-score to reduce one feature, where,

$$\text{Threshold } (t) = \frac{\text{Maximum tolerable F1score reduction}}{\text{Total number of features}} = \frac{T}{n}$$

Step 4: Obtain the F1- score which gives the smallest number of features ($n_{proposed}$).

Note: If there is no value found for the given condition, we will return the same RFE results.

Step 5: To get the relevant feature subset, use feature importance scores (**i**). Then obtain the best $n_{proposed}$ features as the smallest feature subset with similar performance (**s**).

Note: In RFE, remove the least important feature in **i** one by one in the loop, and recalculate RFE and the new **i** for the remaining feature sets until it reaches $n_{proposed}$ features and obtain the relevant feature subset **s**.

outputs:

The smallest number of features with minimum scoring loss: $n_{proposed}$

Relevant feature subset: **s**

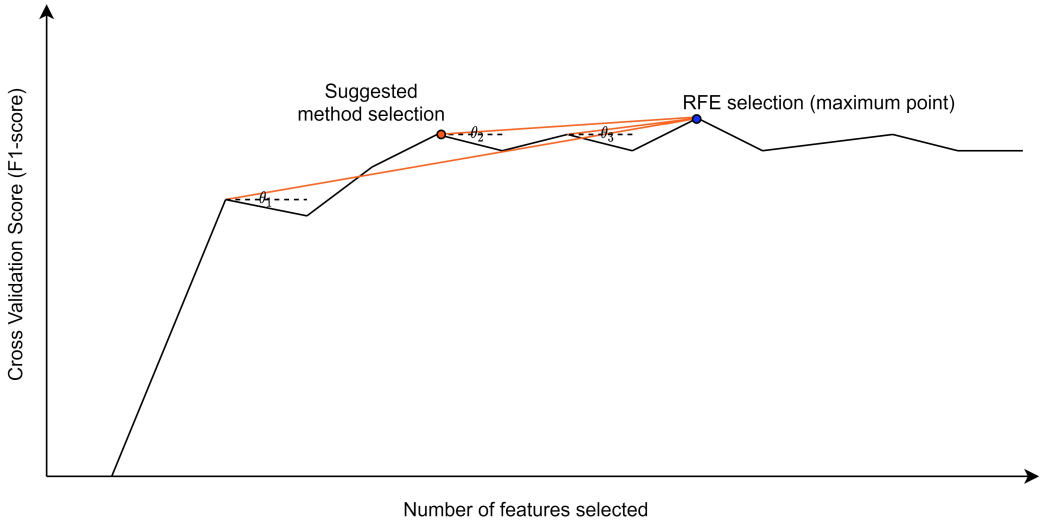

Fig. 3.1. Graphical view of the suggested algorithm. θ_i is the angle between the horizontal dotted line (a line parallel to the number of features selected axis) and the red line, which combines the i^{th} point with the maximum point.

In our algorithm, if we only consider the F1-score that gives the smaller number of features, sometimes we end up with values where the neighbors are larger, and a larger F1-score for the neighbor indicates that the neighbor should be chosen. To avoid such situations and be well-defined, we require the selected value to be a local maximum other than having the smallest number of features.

3.3.4 The role of a threshold

Finding an optimal threshold to distinguish the slight difference between F1-scores was a challenge as it can be depending on many factors. Therefore, when introducing an algorithm, we had to perform a simulation study to determine how the factors affect the behavior of the final result. The gradient method

was introduced to find the F1-score reduction per feature for each selection of subset. We compared each gradient with the maximum bearable gradient value. When we only consider a numerical cut-off value as the threshold, instead of having the gradient method, it will reduce the same amount regardless of the number of features removed. To avoid this problem, the tolerable F1-score should be explained for a single feature reduction. We also observed that when the number of features in the data set increases, the F1-score reduces drastically unless the threshold is extremely small, and it is required to change the threshold according to the number of features in the data set. Therefore, to have consistent solutions, the threshold had to be defined to include the number of features as a parameter. Hence, we considered a tolerable F1-score decrease for one feature, in other words, "the threshold" by having the maximum tolerable F1-score reduction over all features.

3.4 Simulation Study

The results were obtained for different synthetic data sets with a sample size of 1000; the number of features is 30, 50, and 100, such that in each case, half of the features are informative. Results were obtained by fitting five machine learning classification methods on training data. The outputs for three sample data sets were shown in Fig. 3.2, 3.3, and 3.4. The same data set with the imbalance rate of 70%:30% is used for all the different classification models in the same figure. The red points indicate the number of features selected by RFE and the relevant CV F1-score, while the blue points show the smallest practical choice from the new value with a pre-defined threshold and the relevant F1-score. The maximum tolerable F1-score reduction was taken

as 0.05 for all samples. The threshold (F1-score reduction per feature) was calculated accordingly.

For data set 1 (Fig. 3.2), when the number of features in the samples is 30, and the threshold is 0.0017, the proposed method selects seven features instead of 17 with the Logit classifier. Other than the Decision Tree classifier, all the other methods choose smaller feature subsets using the proposed method over the original RFE. For data set 2 (Fig. 3.3), the Logit classifier with RFE selects 36 features, while the proposed method chooses simply 12 features with only 0.0104 reduction of F1-score. The Decision Tree classifier reduces the features from 43 to 6 for an F1-score reduction of 0.0307. For Fig. 3.4, with data set 3 Logit, Decision Tree, and Random Forest classifiers select a lower number of features in the proposed method, while LGBM and Linear SVM solely agreed with the RFE selection under the given threshold.

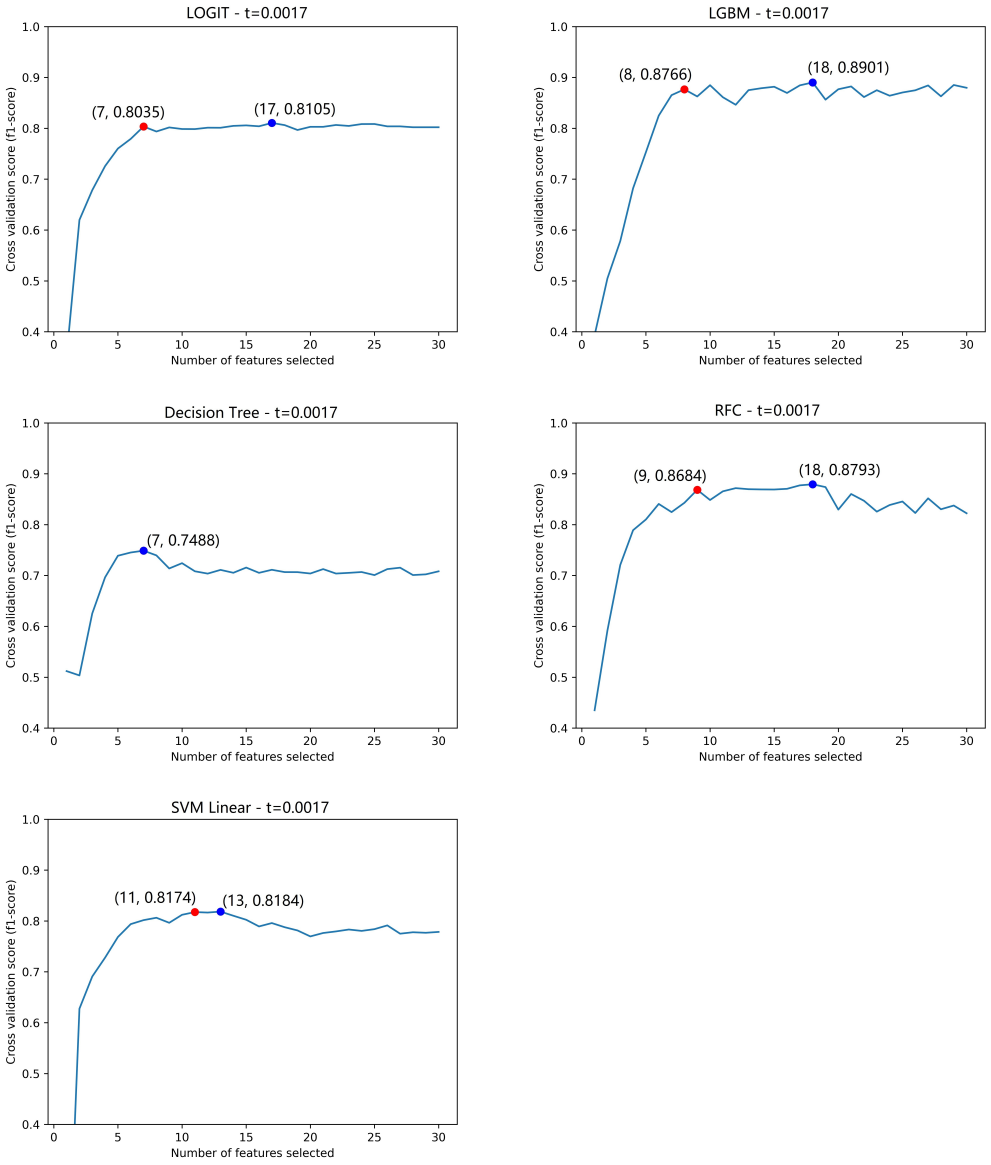

Fig. 3.2. (Data set 1) Selecting a smaller number of features over the RFE selection when the number of features in the samples is 30 and the threshold is 0.0017.

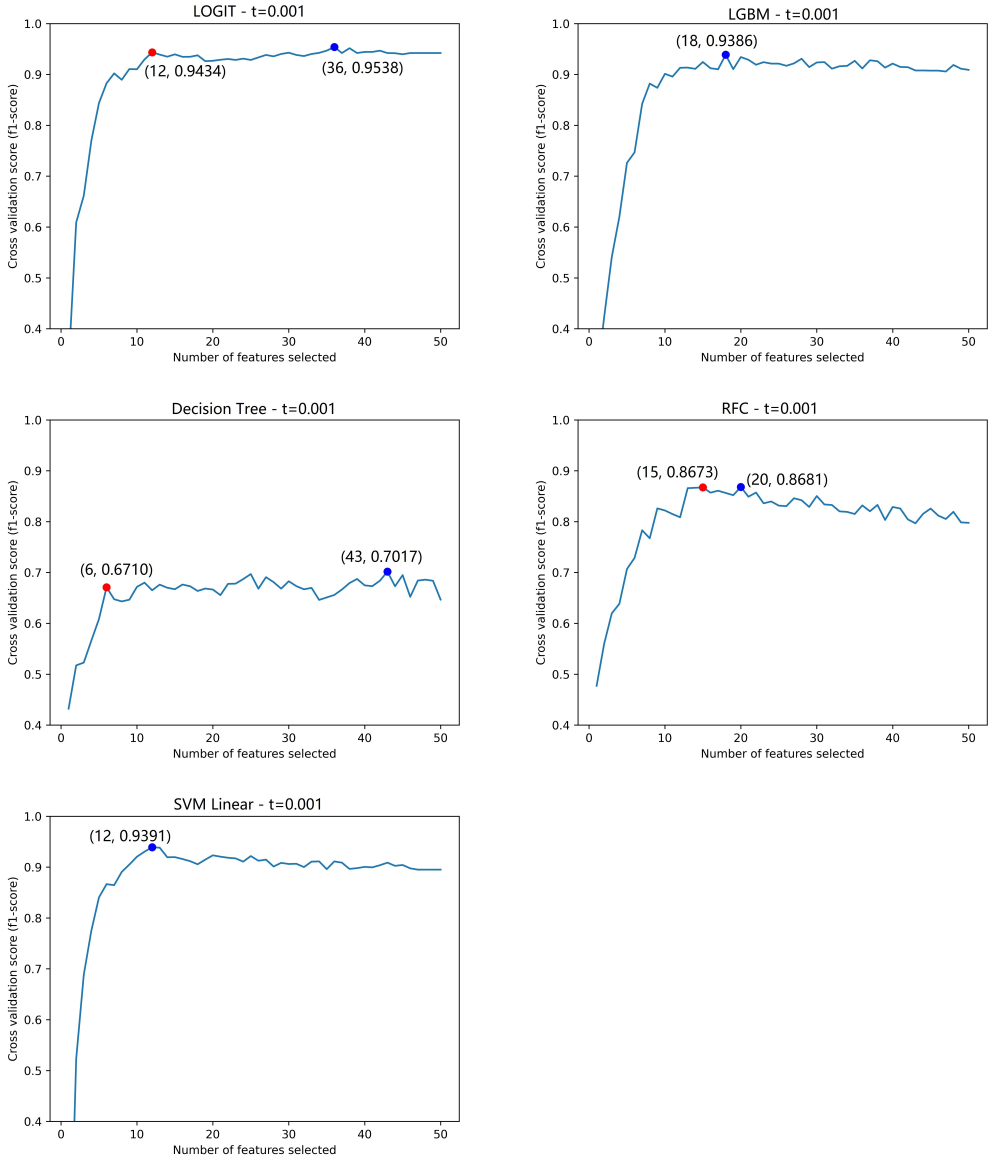

Fig. 3.3. (Data set 2) Selecting a smaller number of features over the RFE selection when the number of features in the samples is 50 and the threshold is 0.001.

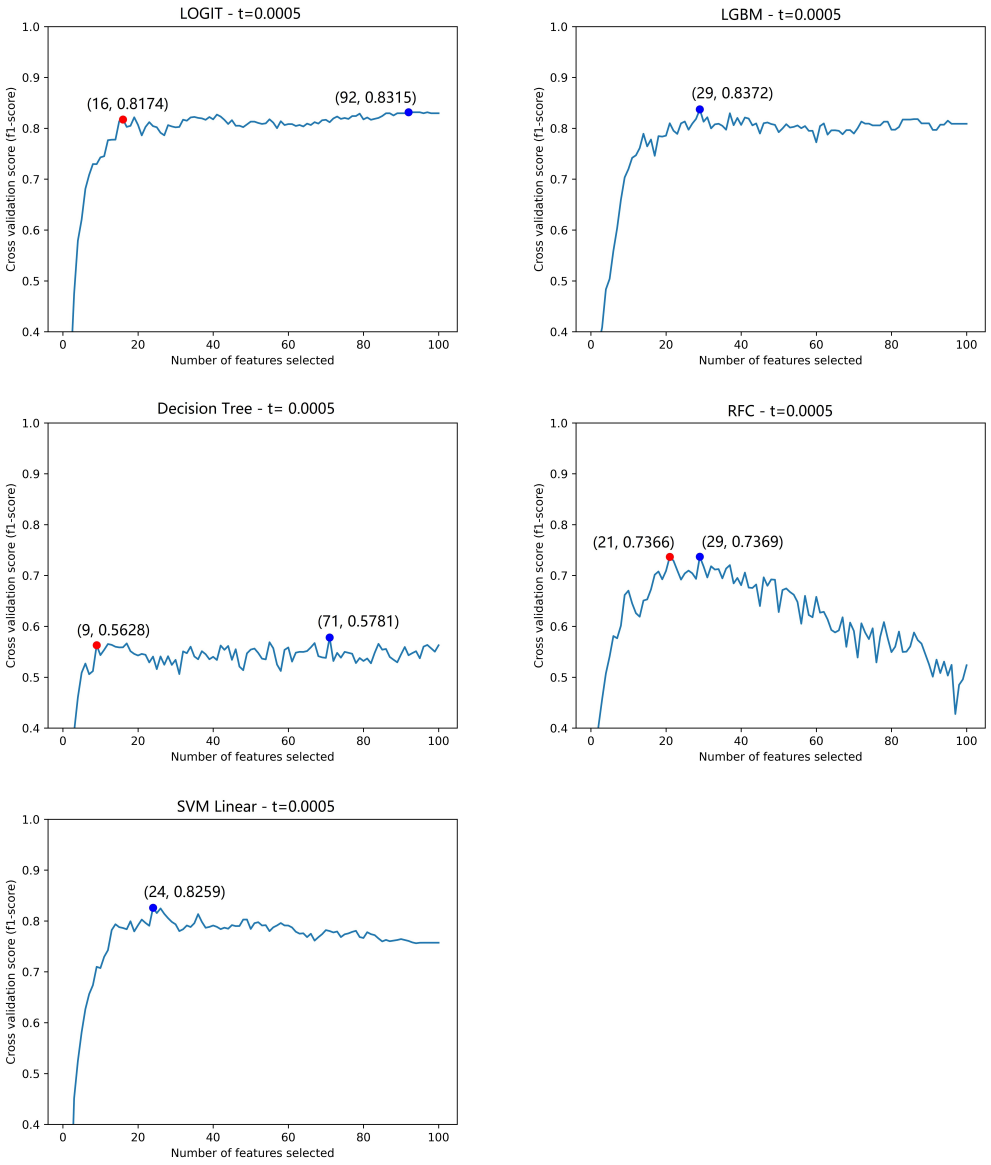

Fig. 3.4. (Data set 3) Selecting a smaller number of features over the RFE selection when the number of features in the samples is 100 and the threshold is 0.0005.

3.4.1 Simulation to derive the importance of the suggested method with different levels of imbalance

We conducted a second experiment to determine the necessity of the suggested approach. One hundred samples are simulated from each scenario to reduce the variability in experimental results, while the number of informative features is increased from 1 to the total number of features. The total number of features consists of informative features and non-informative features. No redundant features or repeated features are included in simulated data sets. We generated data for 50%:50% balanced data and two other imbalance rates, 70%:30% and 90%:10%. Here, we only explained the results of the logistic regression model.

The average number of selected features, the average F1-scores for both RFE and the suggested method were plotted against the number of informative features in the data sets. The comparison is shown in Fig. 3.5, 3.6, and 3.7 for each imbalance rate. Fig. 3.8 presents the minuscule reduction of the F1-score for the proposed method in each situation. For all cases, the maximum tolerable F1-score reduction for the total number of features was taken as 0.05, and the threshold was calculated accordingly.

When the number of informative features increases compare to the number of features in the data set, we can see a notable increase in the feature reduction, especially for more balanced data sets. If we examine all figures, we do not observe a considerable change in the F1-score as the number of features in the data set increases. It is expected since we already incorporated the number of features in the algorithm.

However, it can be seen that the selection of the threshold is susceptible to the class-imbalance rate. As shown in the figures, we are required to use a

Table 3.1: Mean comparison of simulation results

number of features (t)	imbalance rate	mean (std) of number of selected features - RFE	mean (std) of number of selected features - Proposed	mean (std) of CV F1-scores - RFE	mean (std) of CV F1-scores - Proposed
30 (0.0017)	50:50	13.238 (6.511)	9.736 (4.432)	0.935 (0.007)	0.932 (0.007)
	70:30	13.525 (6.514)	10.429 (5.025)	0.902 (0.009)	0.899 (0.009)
	90:10	13.214 (5.798)	11.263 (5.265)	0.819 (0.016)	0.817 (0.016)
50 (0.0010)	50:50	20.311 (11.268)	14.778 (7.063)	0.930 (0.007)	0.926 (0.009)
	70:30	20.416 (11.099)	16.09 (8.228)	0.896 (0.009)	0.894 (0.011)
	90:10	20.143 (9.967)	17.181 (8.552)	0.803 (0.018)	0.801 (0.018)
100 (0.0005)	50:50	38.323 (23.702)	26.631 (14.268)	0.914 (0.012)	0.910 (0.016)
	70:30	39.008 (23.875)	29.850 (16.992)	0.871 (0.018)	0.868 (0.020)
	90:10	38.856 (22.169)	33.273 (18.989)	0.758 (0.032)	0.756 (0.033)

relatively larger threshold value for highly imbalanced data to obtain similar results as balanced data. This sensitivity will be undetectable when applying a re-sampling technique such as Synthetic Minority Over-sampling Technique (SMOTE) on data. In Section 3.5, we obtain results in both scenarios with and without SMOTE for the practical situation.

Table 3.1 reveals how the means and standard deviations vary for the number of selected features and the relevant F1-score. For a smaller reduction of the mean F1-score, we could achieve a higher return from the mean number of selected features in each situation.

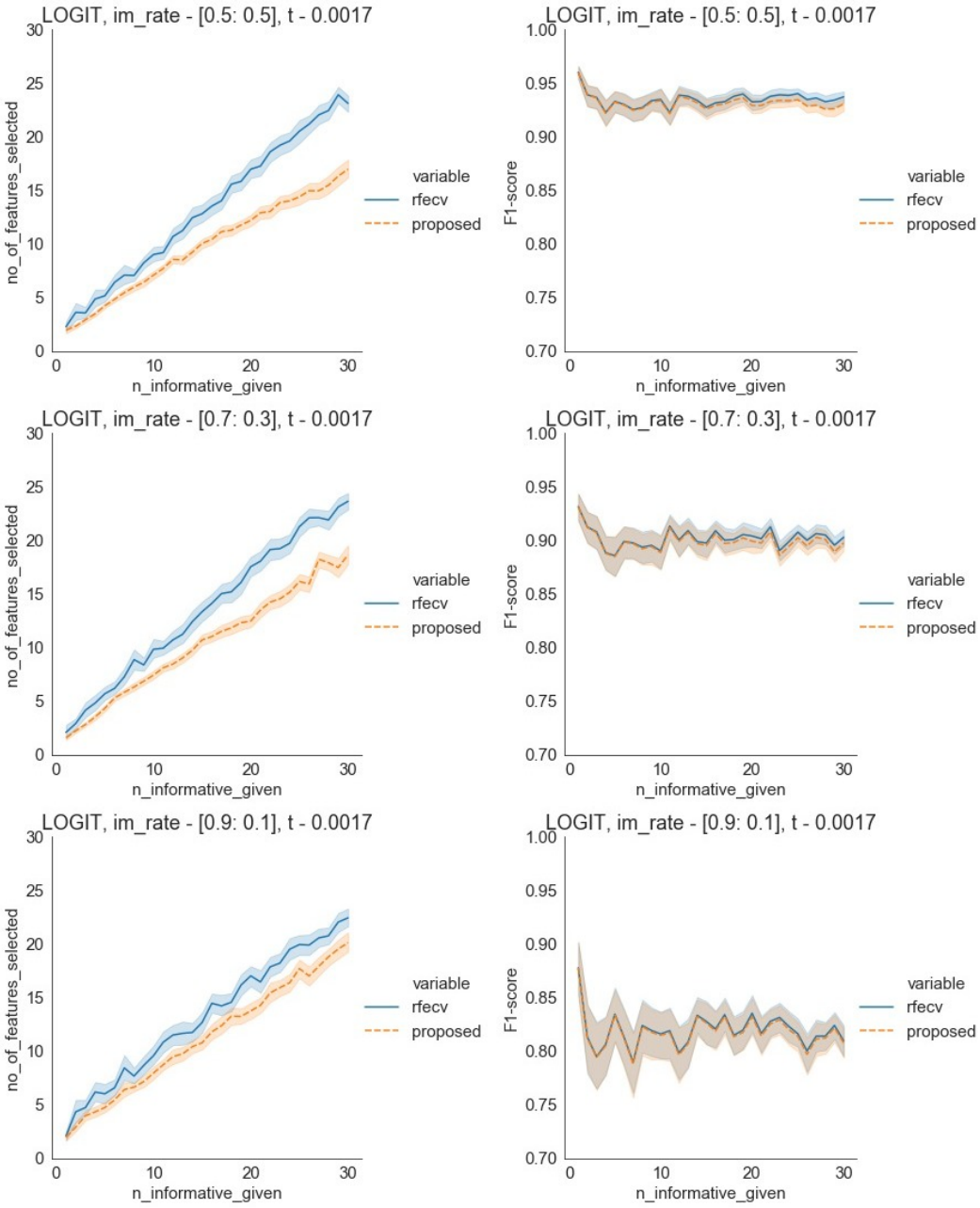

Fig. 3.5. The comparison of the average number of selected features (1^{st} column) and the average CV F1-scores for the two methods (2^{nd} column) for Logit model with 30 features. Each row indicates the imbalance rates 50:50, 70:30, and 90:10, respectively.

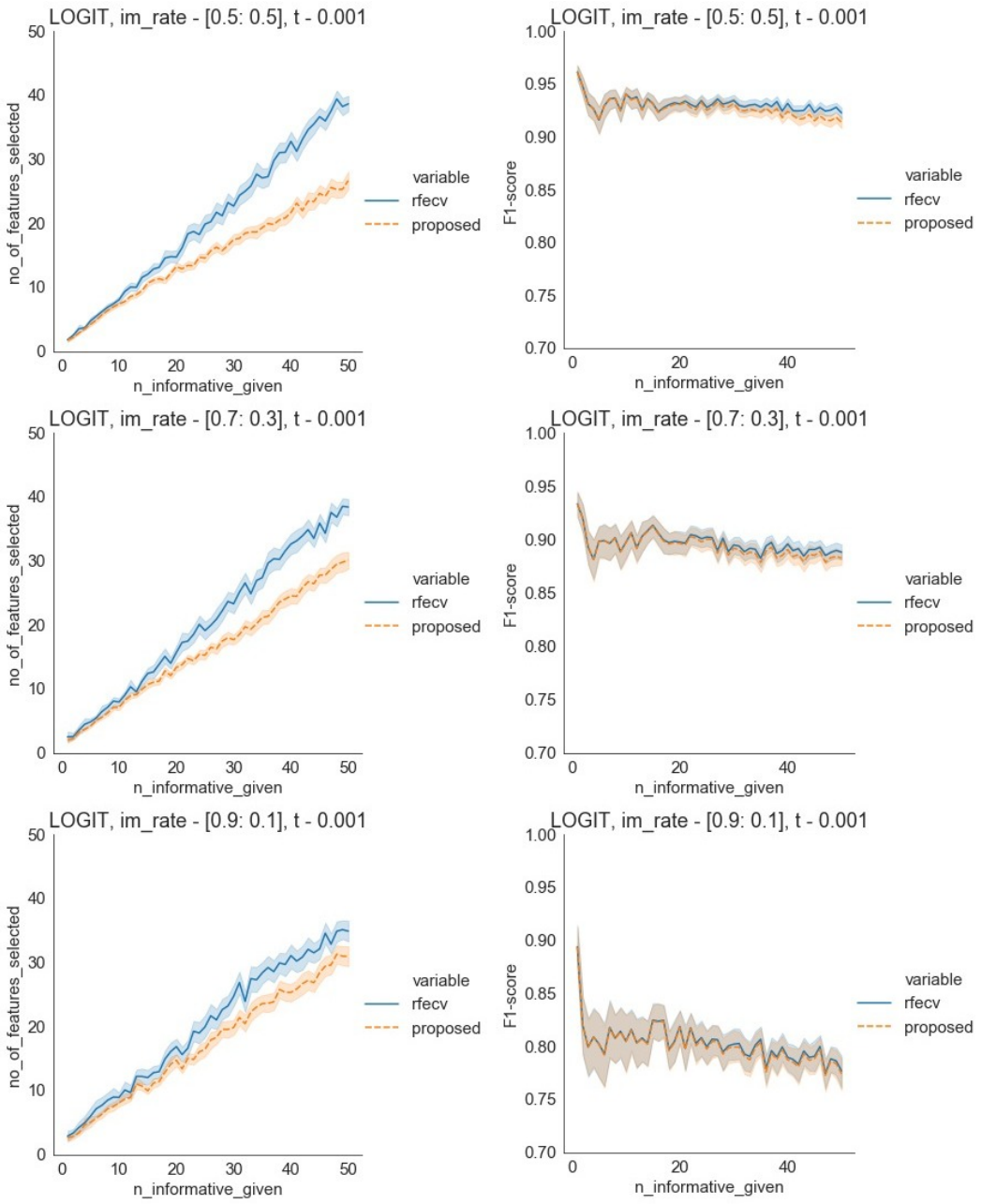

Fig. 3.6. The comparison of the average number of selected features (1^{st} column) and the average CV F1-scores for the two methods (2^{nd} column) for Logit model with 50 features. Each row indicates the imbalance rates 50:50, 70:30, and 90:10, respectively.

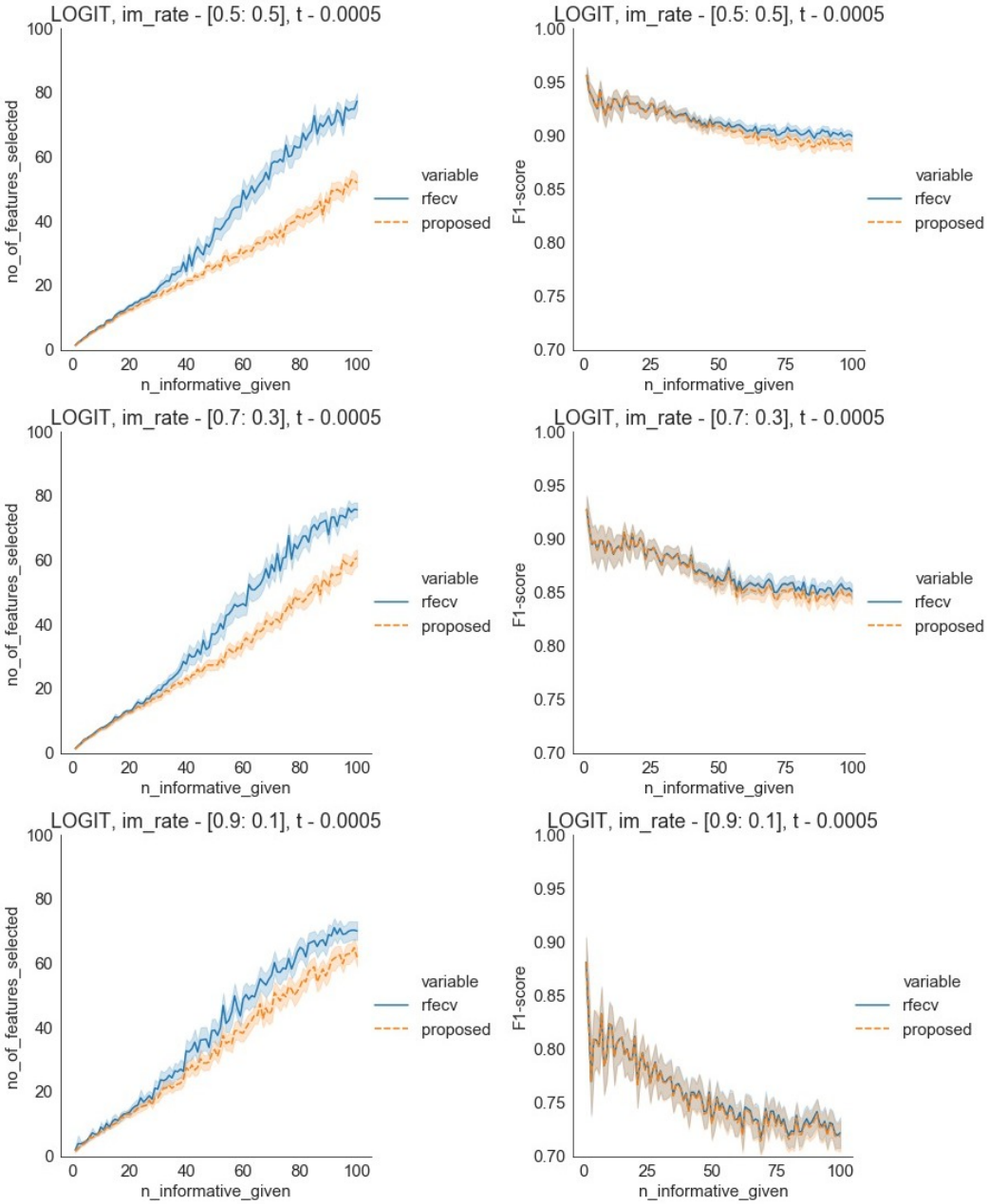

Fig. 3.7. The comparison of the average number of selected features (1^{st} column) and the average CV F1-scores for the two methods (2^{nd} column) for Logit model with 100 features. Each row indicates the imbalance rates 50:50, 70:30, and 90:10, respectively.

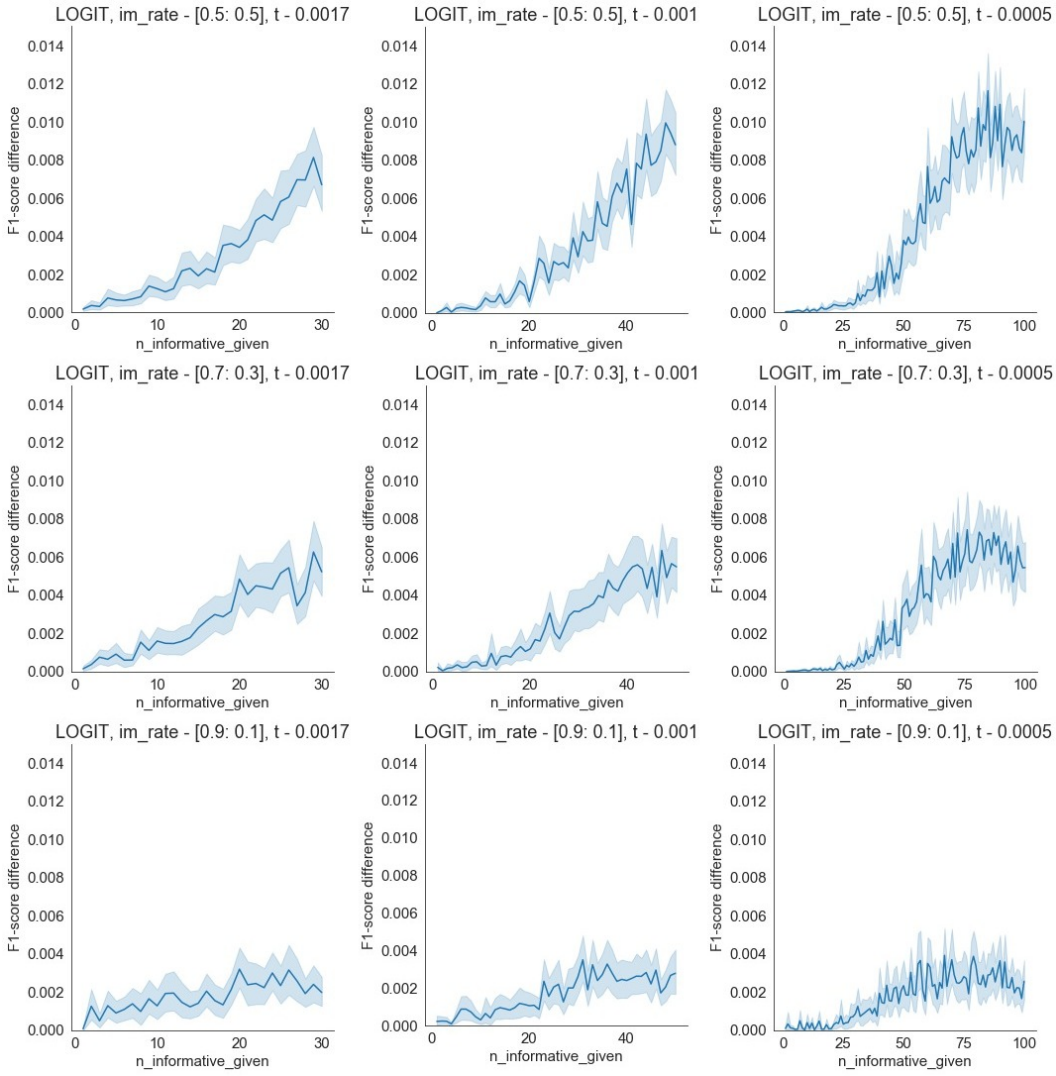

Fig. 3.8. The difference between the two F1-scores for Logit model with 30, 50, and 100 number of features (in columns respectively). Each row indicates the imbalance rates.

Table 3.2: Wilcoxon signed-rank test results for not rejecting the null hypothesis

n_total_features	imbalance_rate	n_informative	p_value	decision
30	[0.9, 0.1]	1	0.054405	no evidence to reject H0
50	[0.5, 0.5]	1	*	*
50	[0.9, 0.1]	1	0.089856	no evidence to reject H0
100	[0.5, 0.5]	1	*	*
100	[0.5, 0.5]	2	0.07865	no evidence to reject H0
100	[0.5, 0.5]	3	0.051235	no evidence to reject H0
100	[0.5, 0.5]	8	0.07865	no evidence to reject H0
100	[0.7, 0.3]	1	*	*
100	[0.7, 0.3]	2	0.054405	no evidence to reject H0
100	[0.7, 0.3]	4	0.089856	no evidence to reject H0
100	[0.9, 0.1]	1	0.089856	no evidence to reject H0
100	[0.9, 0.1]	9	0.07865	no evidence to reject H0
100	[0.9, 0.1]	11	0.158655	no evidence to reject H0
100	[0.9, 0.1]	13	0.07865	no evidence to reject H0
100	[0.9, 0.1]	16	0.054405	no evidence to reject H0
100	[0.9, 0.1]	19	0.051235	no evidence to reject H0

Statistical tests were then conducted on each combination of imbalance rate and the total number of informative features to assess whether the population medians of the number of features selected by two methods differ. Since the distributions and differences are not normally distributed (according to the Shapiro Wilk test), the non-parametric Wilcoxon signed-rank test (Wilcoxon, 1945) was used. It tests the null hypothesis that two related paired samples come from the same distribution with the same medians versus the alternative hypothesis, i.e., two samples come from different distributions where the proposed method selection has a lesser median than RFE. According to the

p-values of the tests for each scenario, out of 540 tests, 524 tests rejected the null hypothesis. Details of the test which could not reject the null hypotheses are shown in Table 3.2. The asterisks (∗) represent some tests that were not conducted due to a lack of grouped data. However, apart from the results shown in Table 3.2, we reject the null hypotheses for all the other combinations concluding that the population median number of features selected by the proposed approach is lesser than the median number of features selected by the Logit-RFE method.

3.5 Application

3.5.1 Churn data

For a company, the loss of clients or customers to competitors is known as customer churn (Huang et al., 2012). It is one of the most critical customer relationship issues, directly impacting the profitability of a company as it is much more challenging in businesses to attract new customers than to retain an existing one as acquiring a new customer is five to six times more costly than retaining an existing one (Hadden et al., 2007). Hence, churn prediction and mitigation have become necessary aspects of business growth. In reality, churn data is highly imbalanced, large-scaled, and high-dimensional. Therefore, feature selection methods have to be applied to test data before the prediction to avoid the dimensionality issue.

In this section, we use the proposed method to discover a solution to this problem. The "telco customer churn" data set, downloaded from IBM community website (IBM, 2019) was used to compare the results of the original

RFE feature selection with the proposed method. After scaling numerical features and including categorical dummy variables, the data set consists of 7043 instances and 40 features. We divide the data set into two groups, 75% training samples, and 25% test samples. The class-imbalanced rate for the data set is 76%:24%, where the minority class represents the churners.

In this section, five classification models were fitted and applied both feature selection methods: RFE and the proposed method on the models. We fixed the random state of the decision tree and the random forest classifiers to have deterministic and reproducible results. Since the data are imbalanced, we also tried to repeat the procedure using re-sampling methods. In this situation, the Synthetic Minority Oversampling Technique (Chawla et al., 2002) was applied to data before fitting the models to have a balanced data set. We observed the number of features and F1-score reduction by the proposed method in each scenario and obtained the feature subset accordingly.

3.5.2 Obtaining a smaller number of features

We obtained the outputs by considering the maximum tolerable F1-score reduction for 40 features as 0.03. Hence, the threshold, in other words, the tolerable F1-score drop per feature, was 0.00075. The resultant graphs with SMOTE data are shown in Fig. 3.9. If we had more room for further F1-score reduction, the same procedure could be repeated with a larger threshold. Hence the outputs for having the maximum tolerable F1-score reduction as 0.05 is shown in Fig. 3.10. It further reduces the number of features as possible. Both Fig. 3.9 and Fig. 3.10 exhibit how the proposed method works to find the smallest local

maximum with a given threshold.

After applying the original RFE algorithm and the proposed method, the resultant output is shown in table 3.3. Feature reduction percentage compared to the RFE was calculated for each scenario using the below equation.

$$\text{Feature reduction\%} = \left(\frac{\text{Difference between number of selected features}}{\text{No of features selected by RFE}}\right) * 100$$

Table 3.3 depicts the advantage of using SMOTE in an application as it increases the overall F1-score for each classification model. Hence we will compare the performance of the two methods with SMOTE. For the LGBM, with SMOTE, for the original RFE, we could gain a 41.18% feature reduction by diminishing 0.0087 of the F1-score with the threshold of 0.00075. Similarly, with the threshold of 0.00125, we could achieve a 50% feature reduction compared to the RFE for a 0.0134 loss of the F1-score. The overall feature reduction percentage in this scenario is 57.5%.

3.5.3 Obtaining the relevant feature subset

When it comes to the churn prediction, it is more important to accurately know the highly relevant features as there will be a hundred other features that could relate to churn data. We used the feature importance attribute facilitated in each classification model to identify the smallest feature subset with similar performance. The selected feature subsets for the LGBM, with SMOTE for RFE and the proposed method with two different thresholds, are compared in table 4.6. Note that the selected features are strict subsets of each other as the same feature ranking order is used for all the subsets.

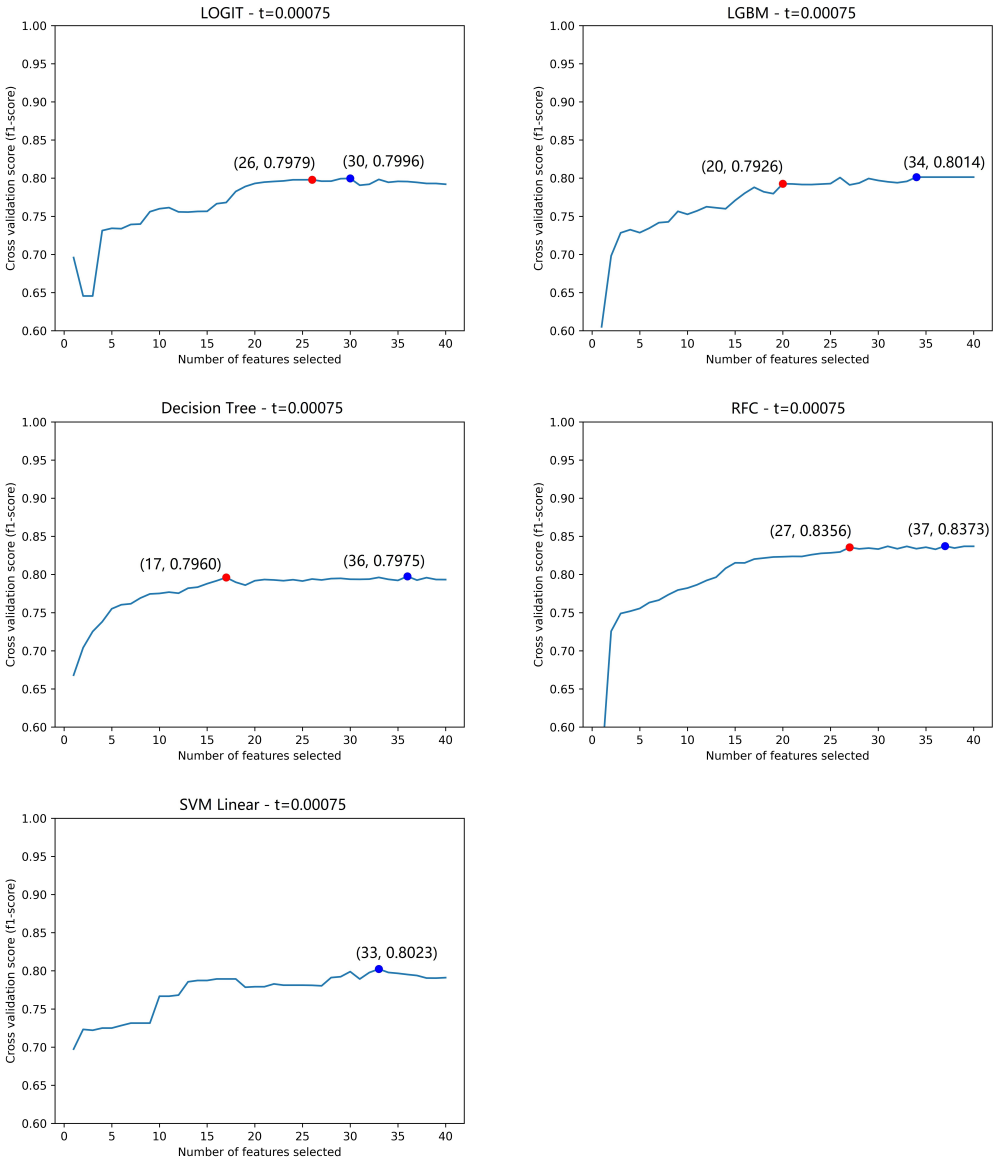

Fig. 3.9. Selecting smaller number of features under the threshold of 0.00075. Blue point indicates the RFE selection while Red point indicates the proposed method selection.

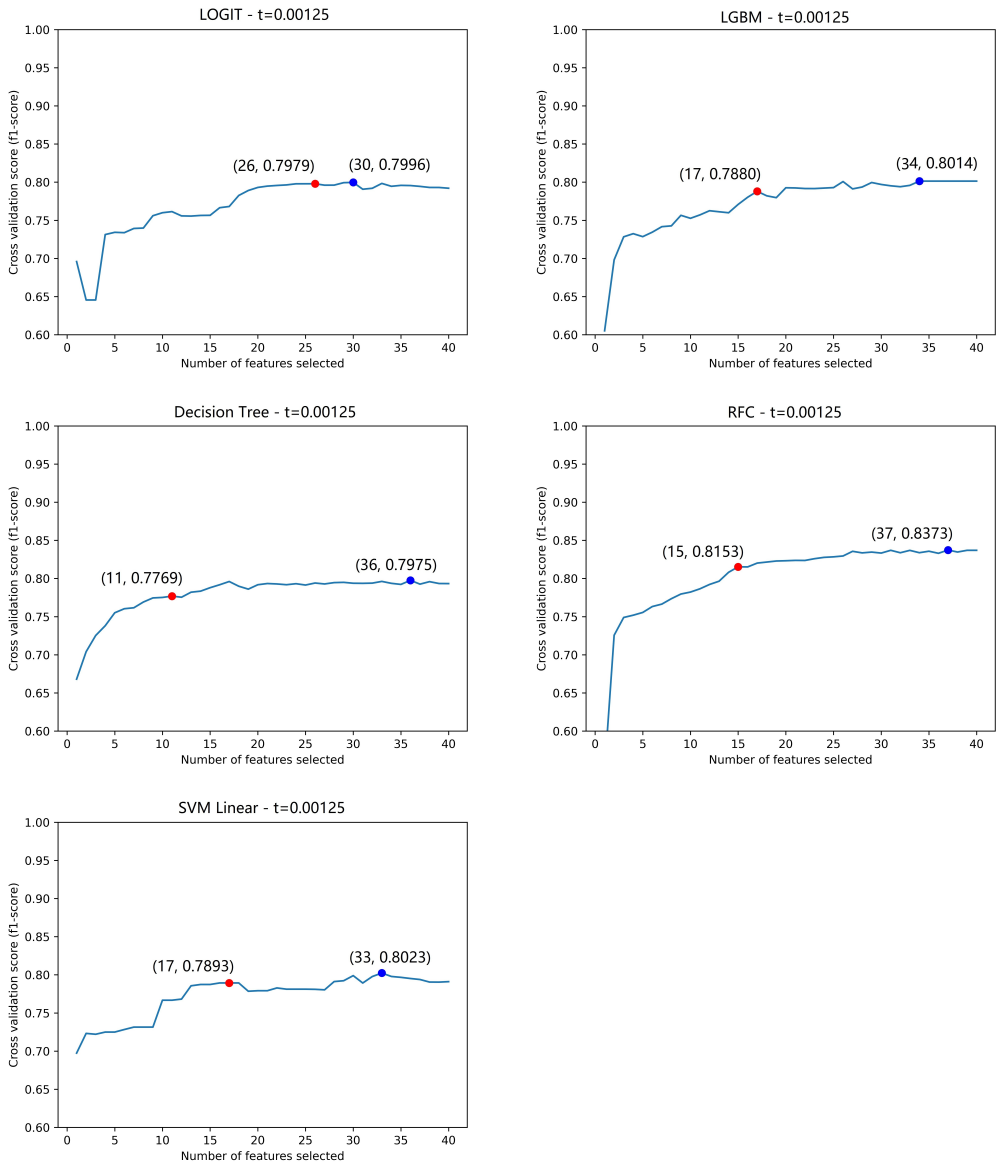

Fig. 3.10. Selecting smaller number of features under the threshold of 0.00125. Blue point indicates the RFE selection while Red point indicates the proposed method selection.

Table 3.3: Feature selection methods comparison for Telco data

SMOTE	Threshold	Model	RFE selection	Proposed selection	Feature reduction%	F1-score RFE	F1-score proposed	F1-score reduction
TRUE	0.00075	logit	30	26	13.33	0.7996	0.7979	0.0018
		lgbm_c	34	20	41.18	0.8014	0.7926	0.0087
		decision tree	36	17	52.78	0.7975	0.7960	0.0015
		rfc	37	27	27.03	0.8373	0.8356	0.0017
		svm linear	33	33	0.00	0.8023	0.8023	-
	0.00125	logit	30	26	13.33	0.7996	0.7979	0.0018
		lgbm_c	34	17	50.00	0.8014	0.7880	0.0134
		decision tree	36	11	69.44	0.7975	0.7769	0.0207
		rfc	37	15	59.46	0.8373	0.8153	0.0220
		svm linear	33	17	48.49	0.8023	0.7893	0.0130
FALSE	0.00075	logit	15	15	0.00	0.5918	0.5918	-
		lgbm_c	31	19	38.71	0.5429	0.5354	0.0075
		decision tree	9	9	0.00	0.5721	0.5721	-
		rfc	32	20	37.50	0.5722	0.5679	0.0043
		svm linear	36	10	72.22	0.5754	0.5747	0.0006
	0.00125	logit	15	4	73.33	0.5918	0.5793	0.0125
		lgbm_c	31	10	67.74	0.5429	0.5229	0.0200
		decision tree	9	9	0.00	0.5721	0.5721	-
		rfc	32	20	37.50	0.5722	0.5679	0.0043
		svm linear	36	10	72.22	0.5754	0.5747	0.0006

Table 3.4: Feature selection of the LGBM classifier from original RFE and Proposed methods (for two different thresholds) with SMOTE data

No	Feature	RFE Subset	Proposed Subsets	
			t=0.00075	t=0.00125
	F1-scores	0.8008	0.7926	0.7880
1	MonthlyCharges	✓	✓	✓
2	tenure	✓	✓	✓
3	TotalCharges	✓	✓	✓
4	gender	✓	✓	✓
5	Partner	✓	✓	✓
6	PaperlessBilling	✓	✓	✓
7	PaymentMethod_Electronic check	✓	✓	✓
8	Dependents	✓	✓	✓
9	SeniorCitizen	✓	✓	✓
10	MultipleLines_No	✓	✓	✓
11	PaymentMethod_Mailed check	✓	✓	✓
12	OnlineBackup_No	✓	✓	✓
13	PaymentMethod_Bank transfer (automatic)	✓	✓	✓
14	OnlineSecurity_Yes	✓	✓	✓
15	PaymentMethod_Credit card (automatic)	✓	✓	✓
16	TechSupport_No	✓	✓	✓
17	DeviceProtection_Yes	✓	✓	✓
18	DeviceProtection_No	✓	✓	
19	MultipleLines_Yes	✓	✓	
20	Contract_One year	✓	✓	
21	TechSupport_Yes	✓		
22	OnlineBackup_Yes	✓		
23	StreamingTV_No	✓		
24	OnlineSecurity_No	✓		
25	StreamingMovies_No	✓		
26	StreamingTV_Yes	✓		
27	StreamingMovies_Yes	✓		
28	Contract_Two year	✓		
29	Contract_Month to month	✓		
30	InternetService_Fiber optic	✓		
31	InternetService_DSL	✓		
32	PhoneService	✓		
33	InternetService_No	✓		
34	MultipleLines_No phone service	✓		
35	DeviceProtection_No internet service			
36	OnlineBackup_No internet service			
37	OnlineSecurity_No internet service			
38	StreamingMovies_No internet service			
39	StreamingTV_No internet service			
40	TechSupport_No internet service			

Chapter 4

A Unified Approach for Feature Selection

4.1 Chapter Overview

This Chapter examines four existing feature ordering techniques to find the most desirable and the most informative ordering mechanism. Using the results, an improved method is suggested in section 4.3 to extract the most informative feature subset from the data set. The new method uses the sum of absolute values of the first k principal component loadings to order the features where k is a user-defined application-specific value. It also applies a sequential feature selection method to extract the best subset of features. In section 4.4, we further compare the performance of the proposed feature selection method with results from the existing Recursive Feature Elimination (RFE) by simulating data for several practical scenarios with a different number of informative features and different imbalance rates. We also validate the method using a

real-world application on several classification methods, and the results are presented in section 4.5.

4.2 Introduction

Feature selection determines the features which should be included in a model. With the curse of dimensionality (Bellman, 1957) data are becoming increasingly high-dimensional, and feature selection is becoming one of the most critical topics to consider. A perfect feature selection method should choose the most informative features and eliminate the less informative features. Therefore, it should primarily be focused on removing non-informative features from the model (Kuhn, 2013) and achieving higher accuracy with the most informative features. The challenge is that it is computationally difficult, time-consuming, and not practical to compare all the combinations of features to determine which combination achieves the highest accuracy.

Therefore, when we make feature selection before applying the predictive classification models, it decreases computational time and improves model interpretability (Miche et al., 2007b). As we highlighted in Chapter 1, it is also better to estimate fewer parameters and reduce the negative impact of non-informative features.

There are several feature selection techniques in literature, yet they behave differently with varying data sets. Most feature selection techniques may not accurately select all informative features when the class sizes are drastically different, and the data set suffers from several non-informative features. Hence, the resulting output may not be the one anticipated. Therefore, identifying the most suitable feature ordering and feature selection technique is a significant

concern requiring a solidified solution. We focus on examining the impact of the number of informative features and the sample size of imbalanced data sets on feature selecting techniques.

Principal component analysis (PCA) is a statistical technique for reducing the dimensionality of high-dimensional data sets. For a data set $\mathbf{X}$ of dimension n by p, PCA attempts to find linear orthogonal combinations of the columns of $\mathbf{X}$ with maximum variance such that $\sum_i A_i X_i = \mathbf{XA}$. After implementing PCA on the data set, the original features will turn into principal components, linear combinations of the original features. Hence, principal components are not as readable and interpretable as original features. Nevertheless, in this Chapter, we consider the PC loadings, the weights of the features of each linear combination, to avoid this interpretability issue.

Wrapper feature selection methods (Kohavi and John, 1997) are one of the commonly used types of feature selection techniques. They evaluate multiple models, adding and/or removing features to find the optimal combination by maximizing overall model performance. Forward Feature Selection, Backward Feature Elimination, and Recursive Feature Elimination (Guyon et al., 2002) are examples of commonly used wrapper methods. For most wrapper methods, the features are ordered according to the importance as an initial step.

Different classification models select features differently with standard wrapper feature selection techniques, even for the same data set. To verify a new general approach for choosing the most important features in any situation, we train our binary prediction models using five commonly-used classification techniques, i.e., Decision Trees (DT) (Guo et al., 2002; Breiman et al., 1984), Random Forest Classifier (RFC) (Breiman, 2001), Logistic Regression (Logit) (McCullagh and Nelder, 1989; Hastie et al., 2009), Light Gradient Boosting

Method (LGBM) (Friedman, 2001), and Support Vector Machine - linear kernel (SVM_lin) (Cortes and Vapnik, 1995; Xia and Jin, 2008).

These classification models are then evaluated using commonly-used performance measures (e.g., F1-score). Recursive feature elimination (RFE) technique is used as the feature selection mechanism, and the synthetic minority oversampling technique (SMOTE) (Chawla et al., 2002) is used as the class re-balancing technique. To cover practical scenarios, we synthetically generated data using "make classification" library in python scikit-learn.datasets (Pedregosa et al., 2011b). However, to better understand the impact of class re-balancing and feature selection on binary classification models, we attempt to address two research questions: finding the best feature ordering technique and determining which method extracts the best informative feature subset.

4.3 Methods and Experimental Design

We began with a simulation study to examine the effect of informative and non-informative features on feature ordering techniques. We simulated data and fixed the total number of features to be 30. In the data simulation, each class is formed of several Gaussian clusters, each located around the vertices of a hypercube in a subspace of dimension equal to the number of informative features. Informative features are drawn independently from Normal(0, 1) distribution for each cluster and then combined as random linear combinations within each cluster to add covariance (Pedregosa et al., 2011b). In our study, the number of classes was two, and there was only one cluster per class. The remaining non-informative features are filled with random noise. Data sets were generated by increasing the number of informative features from 1 to

the total number of features. Since the number of features is fixed at 30, the remaining features are non-informative.

The sample size and the imbalance rate of the data set were changed according to the problem definition. For model validation purposes, each data set was divided into two parts, training (75%) and testing (25%). Finally, we obtained the accuracy measures for classification methods combined with the imbalance rate, sample size, and the number of informative features given in the data set.

4.3.1 What is the best feature ordering technique?

By ordering features, we mean placing them according to their importance to identify the most informative features. We use four different feature selection methods to compare the feature ordering behavior.

1. **Summation of the absolute values of principal component loadings**

   After applying principal component analysis (PCA) to the data set, we are interested in understanding the relationship of the original variables to the principal components using PC loadings (Dunteman, 1989). The PC loadings are the coefficients of the linear combination of the original variables from which the principal components (PCs) are constructed. In PCA, given a mean-centered data set $\mathbf{X}$ with n sample and p variables, the first k principal components, PC_1, PC_2, $\ldots, PC_k$ respectively are given by the linear combination of the original variables $\underline{X}_1, \underline{X}_2, \ldots, \underline{X}_p$,

$$PC_1 = w_{11}\underline{X}_1 + w_{12}\underline{X}_2 + \ldots + w_{1p}\underline{X}_p$$

$$PC_2 = w_{21}\underline{X}_1 + w_{22}\underline{X}_2 + \ldots + w_{2p}\underline{X}_p$$

$$\vdots$$

$$PC_k = w_{k1}\underline{X}_1 + w_{k2}\underline{X}_2 + \ldots + w_{kp}\underline{X}_p.$$

We then compute the sum of the absolute values of the two PC loadings for each feature and order features accordingly. That is for $\underline{X}_i$, it is $\sum_{j=1}^{k} |w_{ji}|$, where $i = 1, \ldots, p$.

2. **Univariate feature selection (ANOVA F value classification)**

   Analysis of variance (ANOVA) is used to analyze the differences among group means in a sample. The F-test is a statistical test used to compare the factors by decomposing the total variation. For example, in one-way or single-factor ANOVA, statistical significance is tested by comparing the F-test statistic,

   $$F = \frac{\text{variability between groups}}{\text{variability within groups}}$$

   These test results can be used in feature selection by removing features independent of the target variable (Butcher and Smith, 2020). We order features according to F values (p values) to identify the most informative features in our analysis.

3. **Absolute correlation of features with the response variable**

   We consider the point biserial correlation to measure the relationship between a binary variable, Y, and a continuous variable, X. This coefficient also varies between -1 and +1 where 0 implies no correlation. The absolute value of a point biserial correlation coefficient $|r_{bp}|$ describes

the magnitude of the relationship between two variables and uses a t-test with $n-1$ degrees of freedom. If we divide the data set into two groups, according to 0 and 1 in Y, the absolute value of the point-biserial correlation can be expressed in the from,

$$|r_{bp}| = \left| \frac{M_1 - M_0}{S_{n-1}} \sqrt{\frac{n_0 n_1}{n(n-1)}} \right|$$

where,

$$S_{n-1} = \sqrt{\frac{1}{n-1} \sum_{r=1}^{n} (X_i - \bar{X})^2}$$

Here, M_0 and M_1 are means on the continuous variable X for all data points in group 0 and 1 respectively, S_{n-1} is the standard deviation of the metric observations, n_0 and n_1 are number of observations for each group and n is the total number of observations (Lev, 1949; Tate, 1954).

4. **Classification model based feature importance**

We now consider the feature importance, which was directly obtained from the classification model trained. More specifically, we look at two main types of more advanced feature importance; they are:

(a) Feature importance from model coefficients (Tsuruoka et al., 2009): Linear machine learning algorithms fit a model where the prediction is the weighted sum of the original features, and these weights (Coefficients) can be used directly to measure the feature importance. Examples include logistic regression and support vector machine with linear kernel.

(b) Feature importance from decision trees (Breiman et al., 1984): Decision tree algorithms like classification and regression trees (CART)

offer importance scores based on minimizing the criterion used to select split points, like Gini or entropy. Examples of such models include decision trees, random forest, and gradient boosting algorithms.

Classification based feature importance types are explained in detail in in Section 2.5.1.

4.3.2 Which method extracts the best informative feature subset?

After identifying the best feature ordering technique, the next challenge is to obtain the most informative feature subset. To achieve this objective, we suggest a better feature selection technique and compare the results with an existing feature selection technique, RFE, which uses model-based feature importance in the initial step. The suggested method uses the sum of absolute values of the principal component loadings and a sequential search method (Guyon et al., 2003; Peng et al., 2010) to extract the most informative features from the data set. The sequential search usually looks for the optimal feature subset by either adding (or removing) a single feature or a small number of features at a time until the specified criteria are fulfilled (Pudil et al., 1994).

The role of principal component loadings

Principal component analysis (PCA), introduced in 1933 (Hotelling, 1933), has been used in different areas, including the biological, physical, and engineering sciences. The prime purpose of the principal component analysis is to reduce

the dimensionality of a multivariate data set and interpret the results by identifying a smaller number of variables. PCA finds the maximum variance in high-dimensional data and projects it onto a smaller dimensional subspace while retaining most of the information. The method requires that correlations be obtained from variables measured on some continuous scale. Also, it assumes a linear relationship between all variables, and large enough sample sizes are required.

However, the most significant disadvantage of PCA is that our original features will be transformed into principal components after implementing PCA on the data set. The principal components are linear combinations of the original features, which are not as interpretable as original features. Hence, we consider using principal component loadings to interpret features and to identify their importance.

Suggested Method: Principal component loading feature selection method
(PCLFS)

The suggested method employs the first k principal components where k is a user-defined application-specific value. We consider only two principal components for the simulation study, assuming the highest sample variances are accumulated in the first few principal components.

The first step is ordering features using the sum of the first two principal component loadings' absolute values, and the second is selecting the optimal feature subset that obtains the maximum F1-score. Starting from the most informative feature, we add features one by one according to the pre-defined order until all features are added. Hence the total number of subsets will

equal the number of features in the data set. We do testing at each step (i.e., F1-score) and, in the end, obtain the feature subset which gives the maximum F1-score. Fig. 4.1 shows the process of the suggested method.

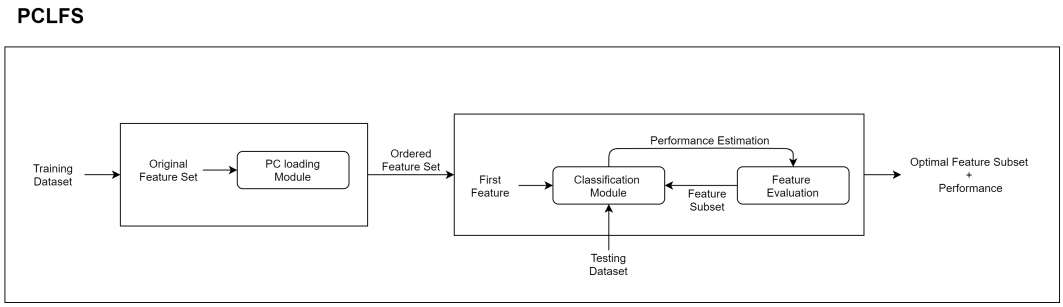

Fig. 4.1. Principal component loading feature selection method

Introducing feature selection confusion matrix

Generally, a confusion matrix is a table that can be used to describe the performance of a classification model. Here, since we already know the number of informative and non-informative features in the simulated data set, we introduce a new confusion matrix to check the feature selection performance of each classification method with any feature selection method.

The new feature selection confusion matrix can be defined as in Table 4.2 where Table 4.1 is the regular model confusion matrix. In the Tables, the outcomes are: P = Positive (Informative), N = Negative (Non-informative), TP = True Positive, FP = False Positive, TN = True Negative and FN = False Negative.

Table 4.1: Model confusion matrix

		Predictions	
		Class 1	Class 0
Actual	Class 1	TP_{model}	FN_{model}
	Class 0	FP_{model}	TN_{model}

Table 4.2: Feature selection confusion matrix

	selected	not selected
informative	TP_{fs}	FN_{fs}
non-informative	FP_{fs}	TN_{fs}

Finally, to evaluate the best performing feature selection method, instead of model F1-score, we also use the feature selection correct percentage (balanced accuracy) obtained using the newly introduced confusion matrix. The feature selection correct percentage can be calculated as below,

$$TPR_{fs} = \frac{TP_{fs}}{\text{Total Number of informative features}}$$
$$FPR_{fs} = \frac{FP_{fs}}{\text{Total Number of informative features}}$$
$$TNR_{fs} = \frac{TN_{fs}}{\text{Total Number of non informative features}}$$
$$FNR_{fs} = \frac{FN_{fs}}{\text{Total Number of noninformative features}}$$
$$Correct\%_{fs} = \frac{TPR_{fs} + TNR_{fs}}{2}$$

Other accuracy measures interpreted based on the newly introduced confusion matrix (Table 4.2) are discussed in Appendix A.1.

To evaluate the model performance in the real-world data set, we also use model Precision and Recall, which can be calculated as

$$\text{Precision} = \frac{TP_{model}}{TP_{model} + FP_{model}}$$
$$\text{Recall} = \frac{TP_{model}}{TP_{model} + FN_{model}}$$
$$\text{F1-score} = 2\left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}\right)$$

4.4 Simulation Results

To find the best performing method, we simulate one hundred different data sets (with a sample size of 1000 and 30 features) for each imbalance rate. The number of informative features was given from 1 to 30, increasing by 1. For the synthetic comparison, we compare feature ordering only using Logistic regression classification.

4.4.1 What is the best feature ordering technique?

The objective of this analysis was to identify the best informative feature ordering techniques. As we are using simulated data, we already know the informative features and non-informative features; hence, four feature ordering techniques were applied to determine which would select the informative features first.

The Fig. 4.2, Fig. 4.3 and Fig. 4.4 show three examples of the ordered selection of the features for different imbalance rates from each method when 20 informative features were included in the data set (indicated by dashed lines). The informative features are labeled as 'i_*'. In contrast, the non-informative features indicate 'n_*'. Irrespective of the imbalance rate, the x-axis of the figures clearly shows that the method of having the summation of the absolute values of the first two principal components loadings identified more informative features than the other methods. The ANOVA F classification and the absolute point biserial correlation order features in a similar manner.

To observe the variability of the results, we repeatedly generated 100 data sets for each scenario to meet different practical situations by changing the sample size, the number of informative features, and the class imbalance rate. We applied all four methods for each data set and finally calculated the percentage of selecting informative features using below equation 4.1.

$$\text{Percentage of informative selected (P)} = \frac{S}{I} \tag{4.1}$$

where S is the average number of informative features selected within the expected range and I is the number of informative features given (i.e., expected range is the total number of informative given in the data set).

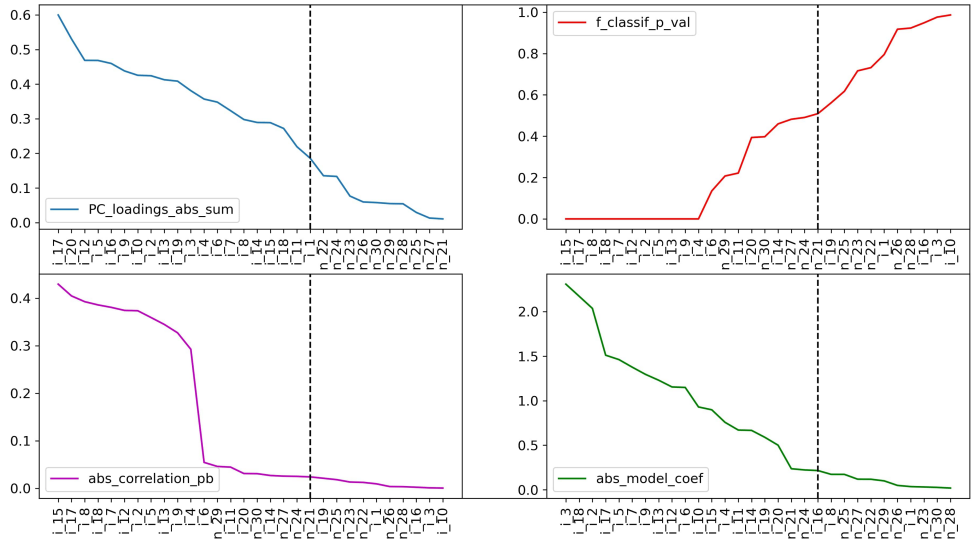

Fig. 4.2. Example result for comparison of four methods for 50%:50% balanced data: X-axis ordered the features according to the importance given by each method.

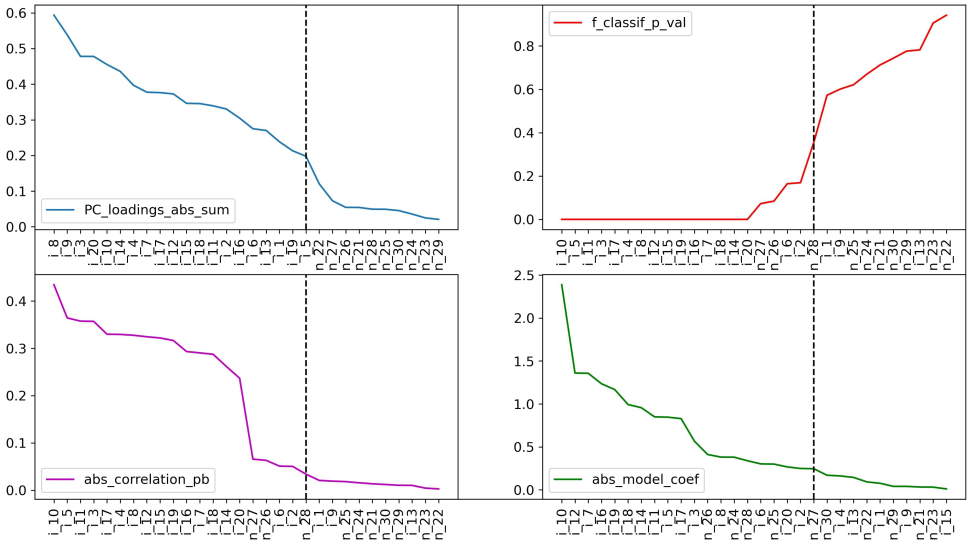

Fig. 4.3. Example result for comparison of four methods for 70%:30% imbalanced data: X-axis ordered the features according to the importance given by each method.

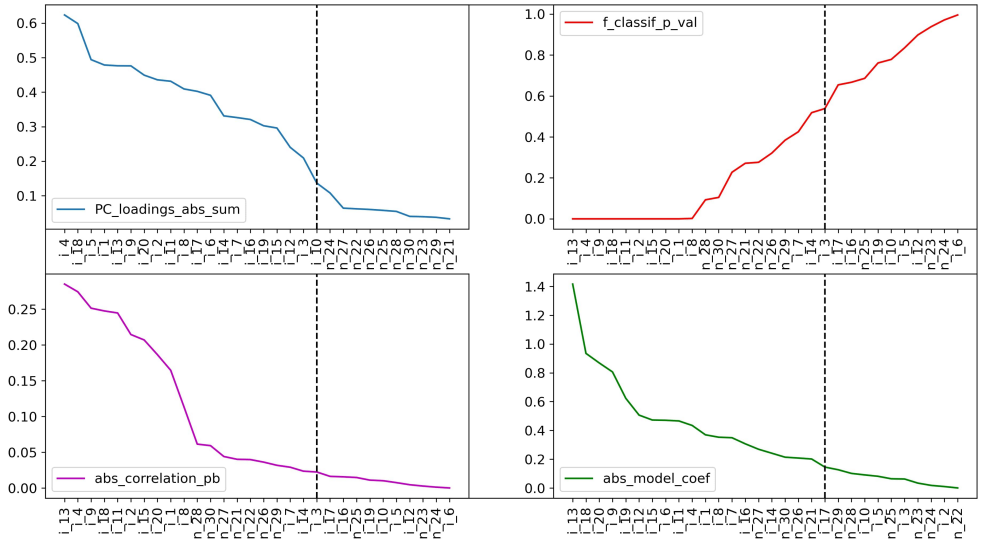

Fig. 4.4. Example result for comparison of four methods for 90%:10% imbalanced data: X-axis ordered the features according to the importance given by each method.

According to Fig. 4.5, Fig. 4.6 and Fig. 4.7, it is noted that until four features, the Logit based method picks the more informative features correctly and when there are more than four informative features in the data set, the sum of absolute principal loading method picks most informative features within the expected range compared with the other three methods. It is also shown that the ANOVA F classification and the absolute point biserial correlation behave identically in all situations.

Since the PC loading method is able to rank the features informatively, there should be a way to extract these informative features in the first phase. Hence, we apply a sequential feature selection technique on the ordered list of features constructed by the PC loading method to obtain the desired feature subset. We choose the Logit absolute coefficient method and recursive feature

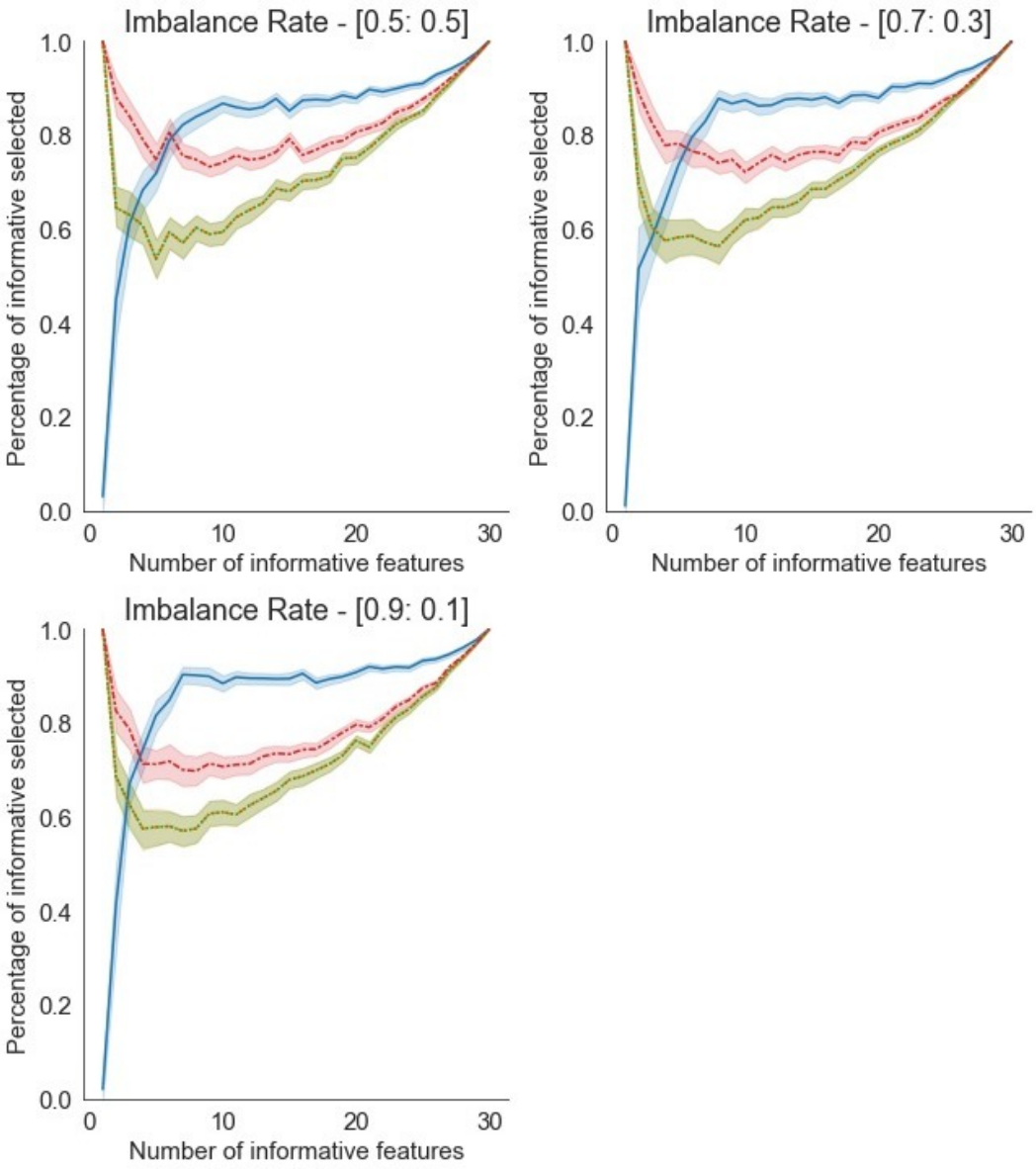

Fig. 4.5. Mean percentages of informative features selected by each ordering technique in different class imbalanced levels with 200 sample sizes. The blue line represents the sum of the absolute values of principal component loadings; the red dashed line indicates Logit model-based feature importance results. The overlapped green and orange dashes lines show the absolute correlation and the ANOVA F value classification results.

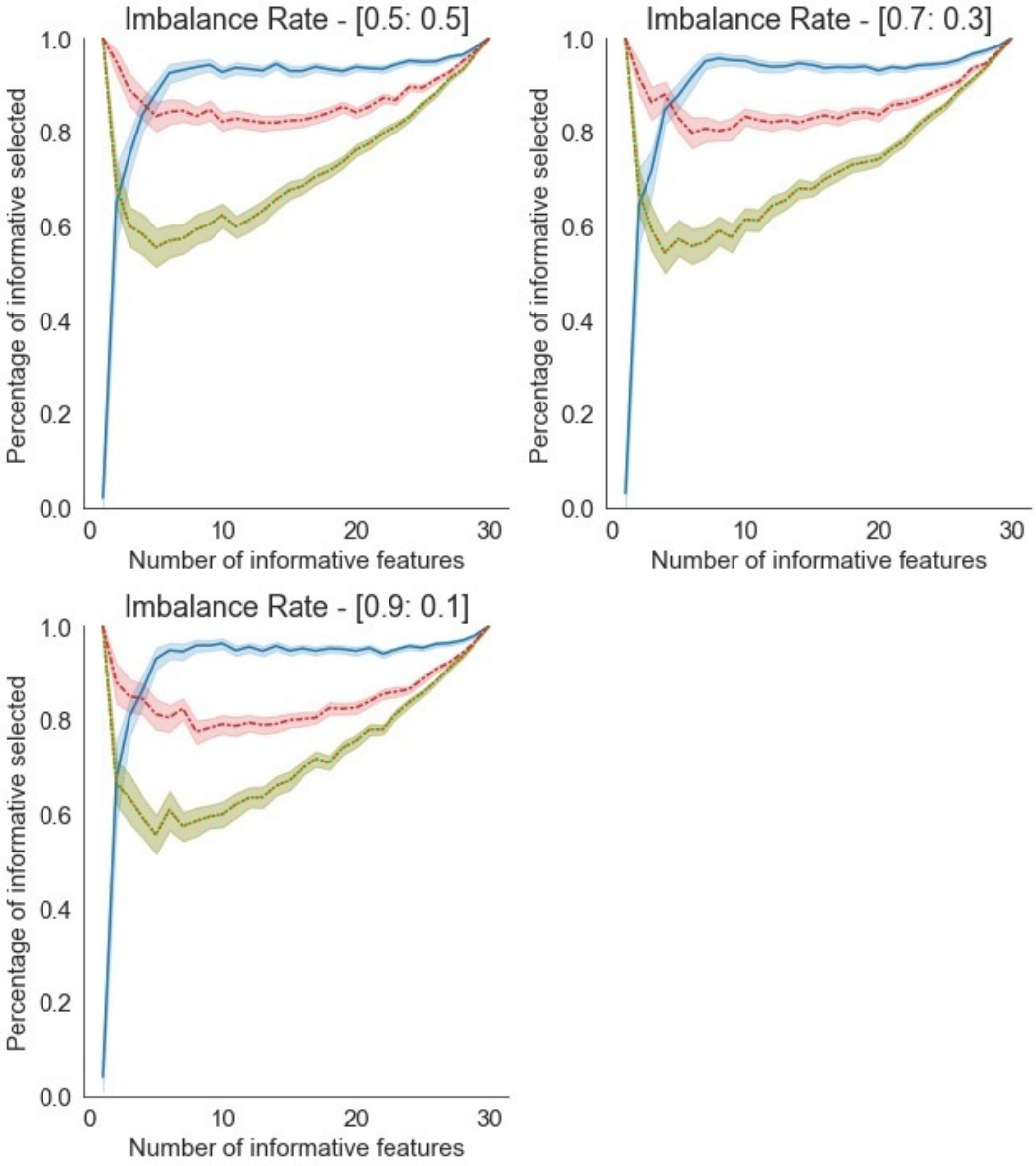

Fig. 4.6. Mean percentages of informative features selected by each ordering technique in different class imbalanced levels with 500 sample sizes. The blue line represents the sum of the absolute values of principal component loadings; the red dashed line indicates Logit model-based feature importance results. The overlapped green and orange dashes lines show the absolute correlation and the ANOVA F value classification results.

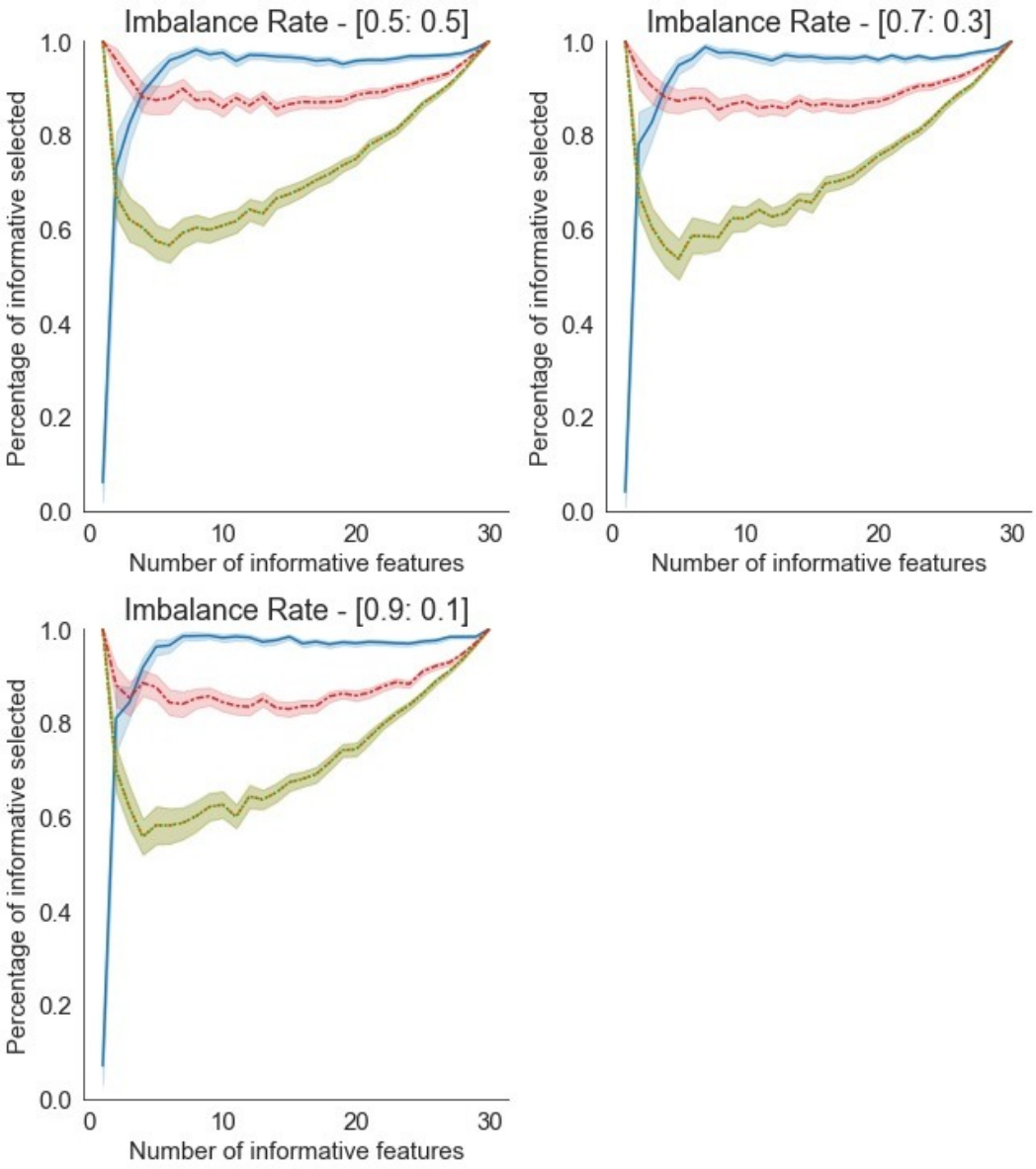

Fig. 4.7. Mean percentages of informative features selected by each ordering technique in different class imbalanced levels with 1000 sample sizes. The blue line represents the sum of the absolute values of principal component loadings; the red dashed line indicates Logit model-based feature importance results. The overlapped green and orange dashes lines show the absolute correlation and the ANOVA F value classification results.

elimination with the Logit classifier to compare the results of the suggested method. This is described in detail in the next section. Results obtained for other classification models, with and without SMOTE are attached in Appendix B.

4.4.2 Which method extracts the best informative feature subset?

To extract the best feature subset, first, we sorted the features using the sum of the absolute values of the first k PC loadings and then obtained the F1-scores for each subset of features by fitting a classification model starting from the most important feature and then adding the next important feature until the least important one. Finally, we compared the results with the existing recursive feature elimination technique, which uses classification model-based feature importance to order data.

Fig. 4.8, Fig. 4.9, Fig. 4.10, and Fig. 4.11 show a comparison of the cross-validation F1-scores for each subset selected by the Logit-RFE and the F1-score of the PCLFS subsets while changing the number of informative features in the data set. The results are for one set of simulated data with sample size 1000, 70%:30% imbalanced rate while increasing the number of informative features from 1 to 30. Points on the lines indicate the maximum F1-scores. In most situations, PCLFS yields a higher F1-score than the Logit-RFE. It is also notable that even by looking at the PCLFS line, we can identify the informative feature count in the data set. Until it reaches the total number of informative features, the F1-score for the PCLFS method increases rapidly, and the line becomes stable afterward.

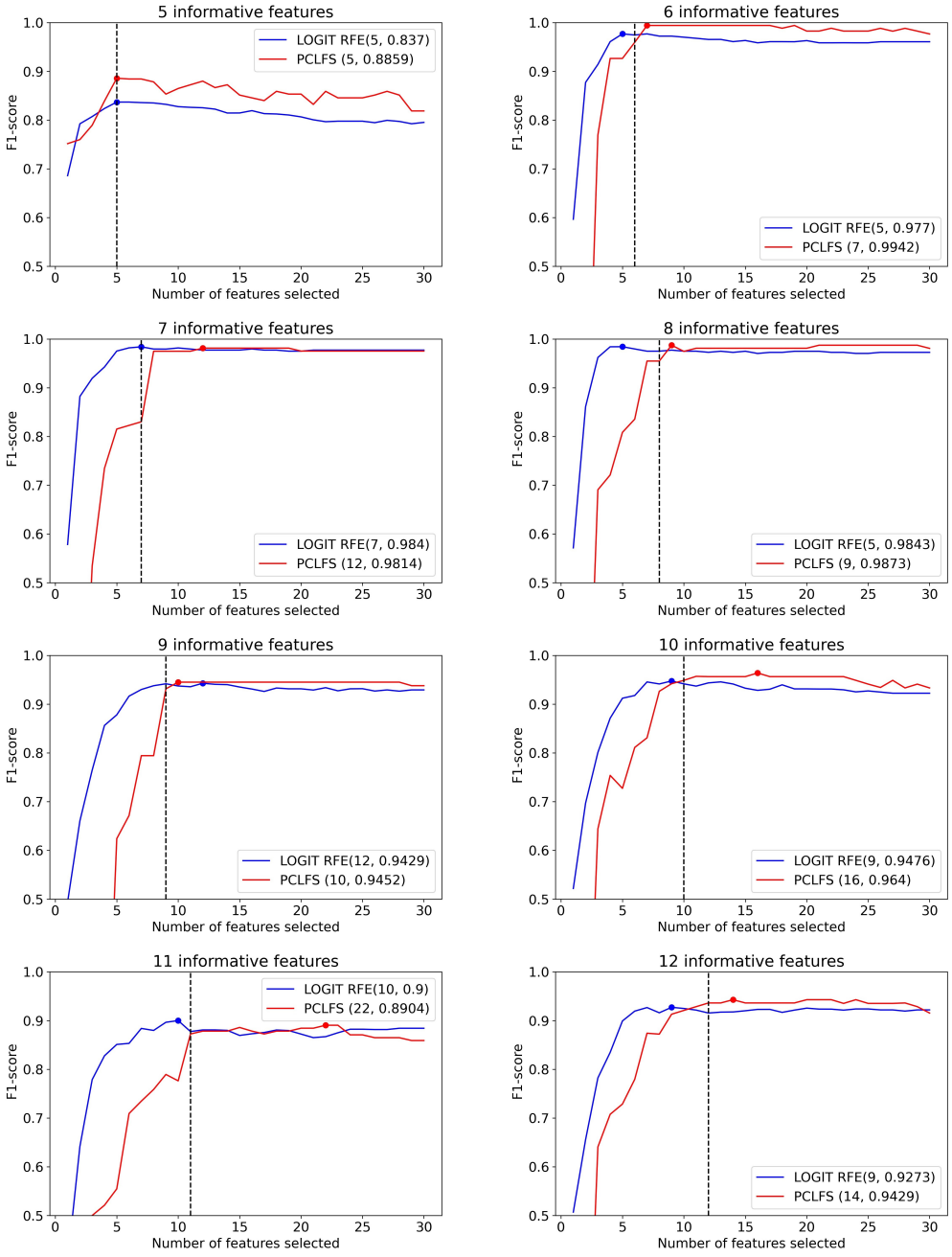

Fig. 4.8. Comparison between Logit-RFE cross validation F1-scores and PCLFS F1-scores with 5-12 informative features

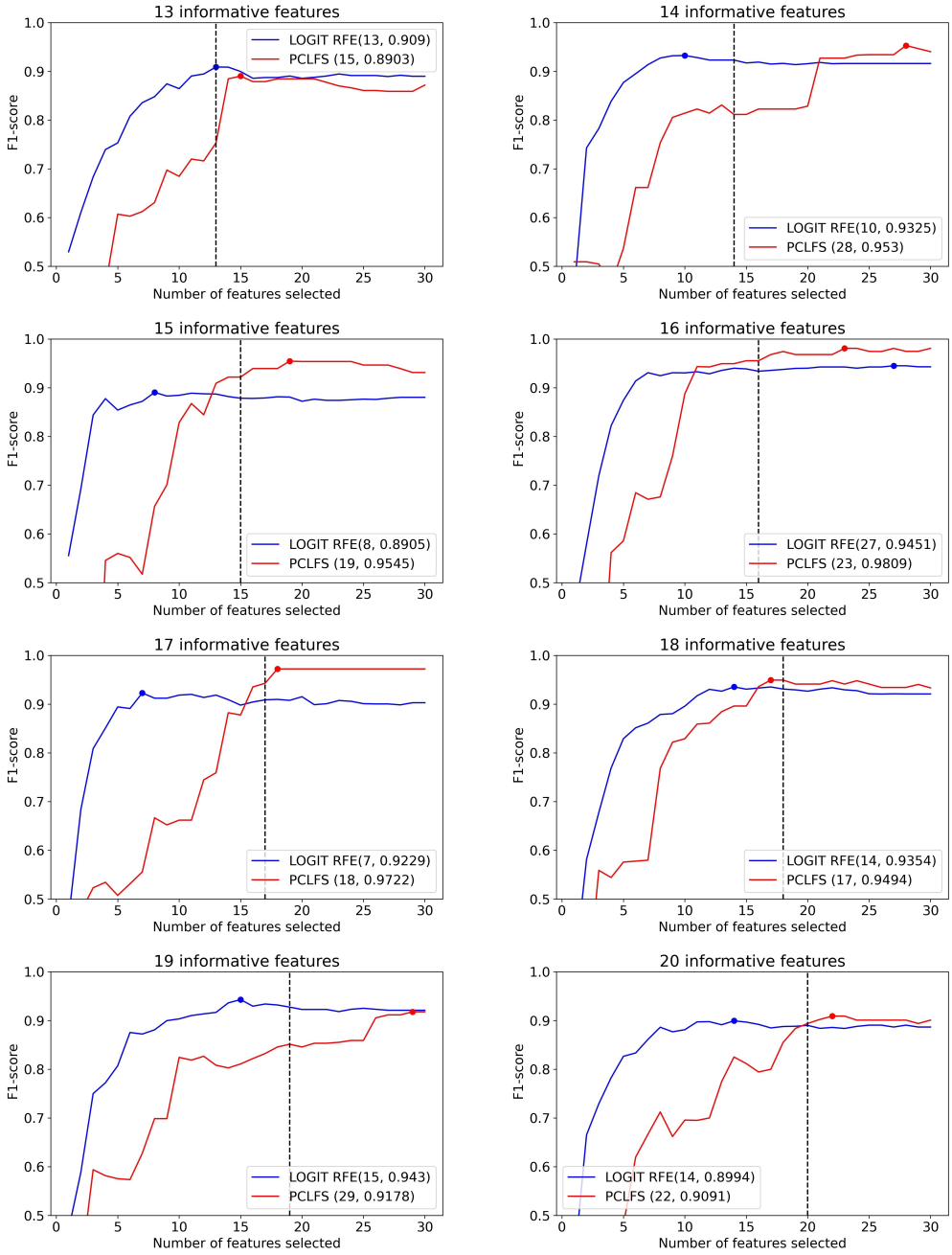

Fig. 4.9. Comparison between Logit-RFE cross validation F1-scores and PCLFS F1-scores with 13-20 informative features

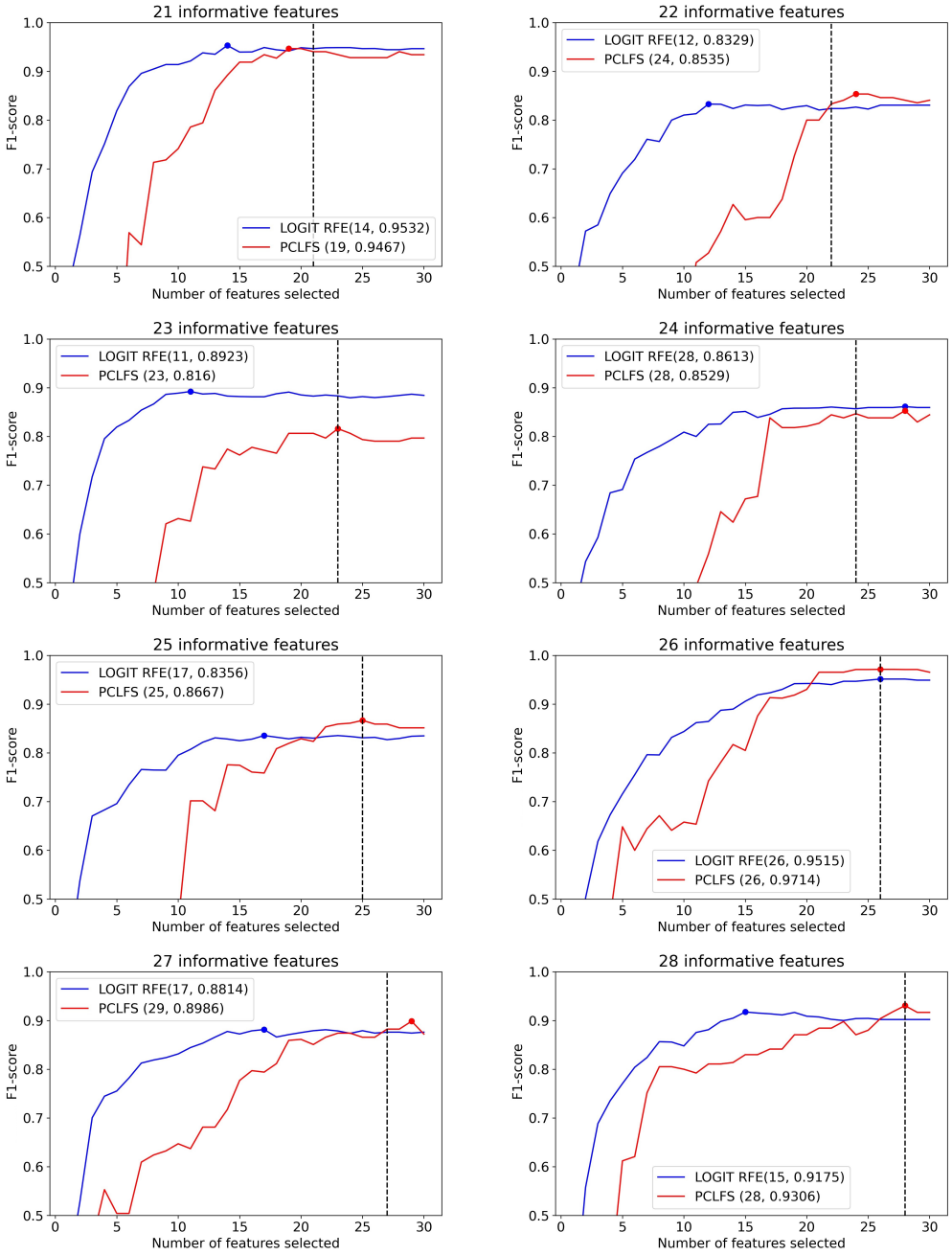

Fig. 4.10. Comparison between Logit-RFE cross validation F1-scores and PCLFS F1-scores with 21-28 informative features

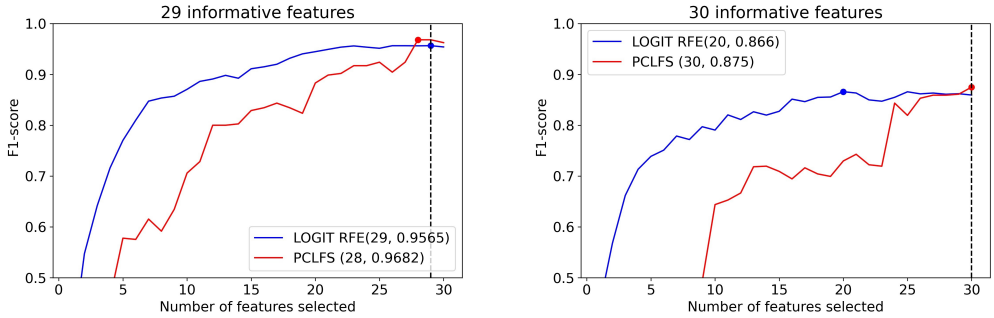

Fig. 4.11. Comparison between Logit-RFE cross validation F1-scores and PCLFS F1-scores with 29-30 informative features

Simulation Results

Again to capture the variability of the results, a simulation study was done by applying both PCLFS and Logit-RFE methods on training data. Then we evaluated the prediction results on testing data and recorded the model F1-score and the feature selection correct percentage (Correct$\%_{\text{fs}}$) in both cases. The process repeated 100 times for three different imbalance rates and three different sample sizes.

According to the Fig. 4.12, Fig. 4.13 and Fig. 4.14, it can be seen that the PCLFS with Logit classifier gives a higher final F1-score in each situation and it works extremely well for small sample size and highly imbalanced data. The figures on the right-hand side imply that the PCLFS method with Logit-classifier obtains a higher feature selection correct percentage in each situation when the number of informative features in the sample is greater than four. Fig. B.8 and B.9 show that with SMOTE, we can eliminate the imbalance and the effect of the imbalance on the F1-score and correct percentages respectively.

Statistical tests were then conducted on each combination of sample size, imbalance rate, and number of informative features to assess whether the

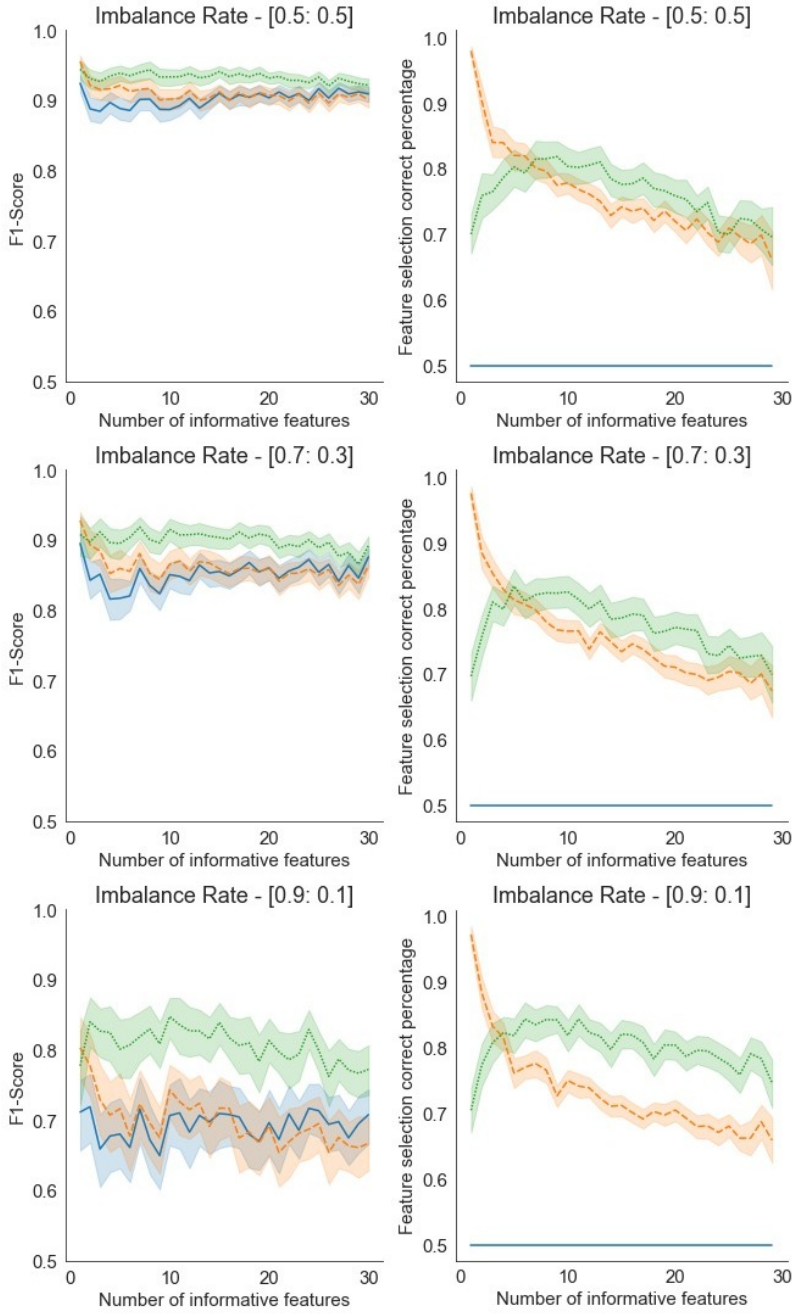

Fig. 4.12. Final model F1-scores and feature selection correct percentages for the Logit model when the sample size is 200. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

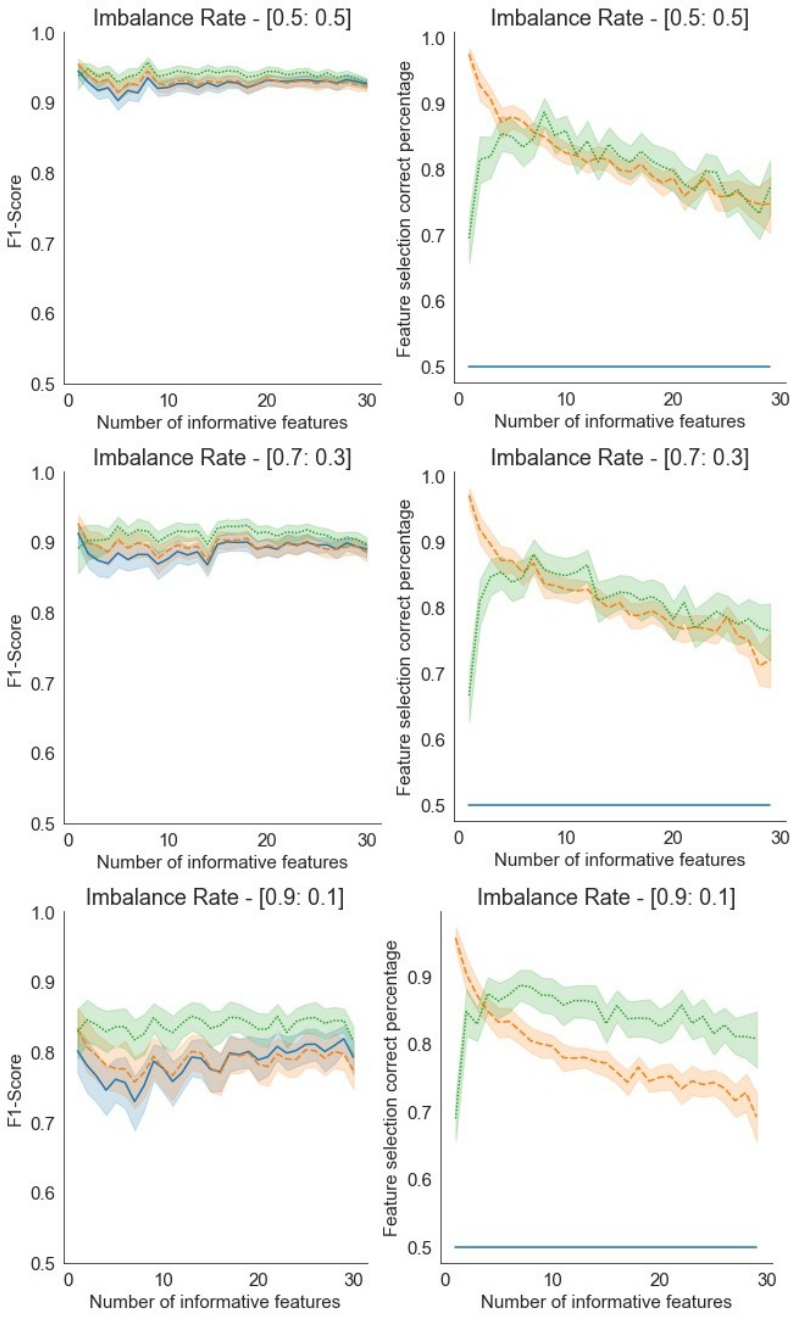

Fig. 4.13. Final model F1-scores and feature selection correct percentages for the Logit model when the sample size is 500. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

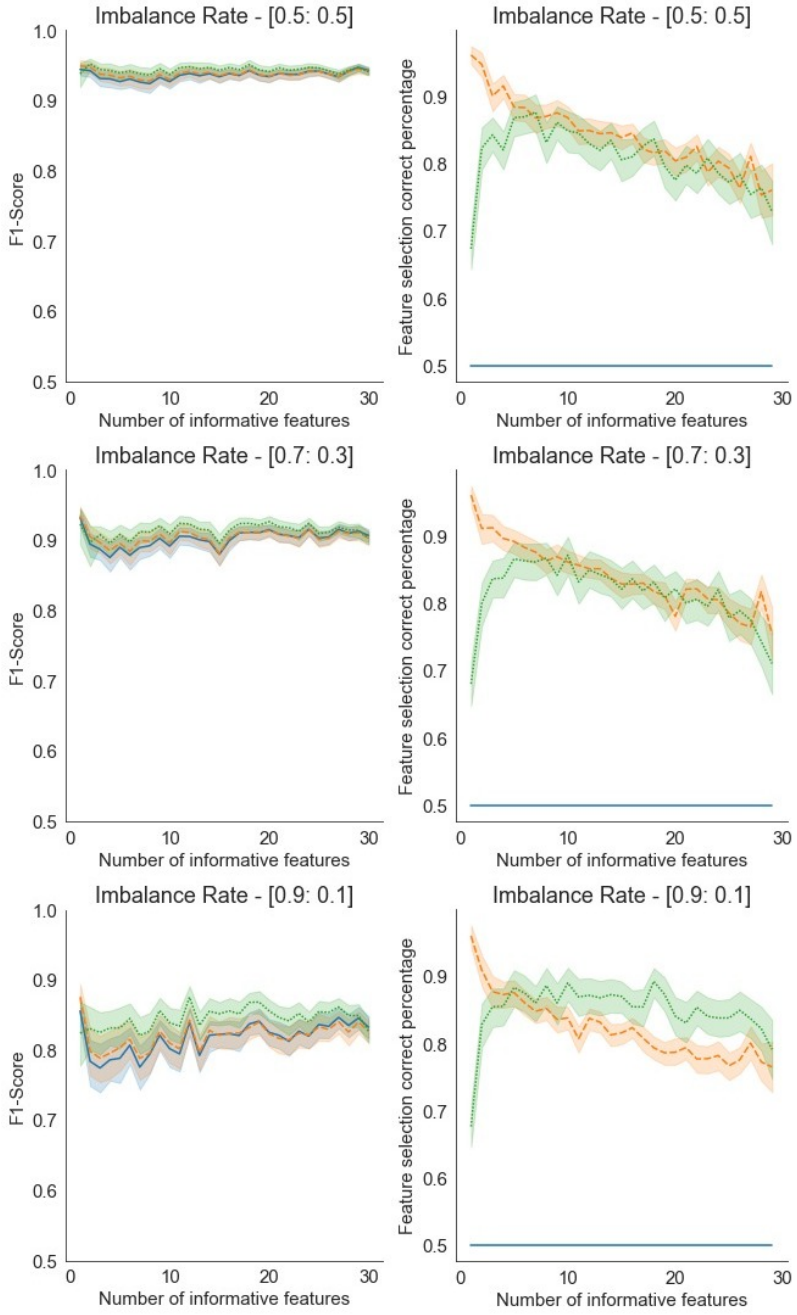

Fig. 4.14. Final model F1-scores and feature selection correct percentages for the Logit model when the sample size is 1000. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

population medians of the F1-scores given by two methods differ. Since the distributions and differences are not normally distributed (according to the Shapiro Wilk test), the non-parametric Wilcoxon signed-rank test (Wilcoxon, 1945) was used. It tests the null hypothesis that two related paired samples come from the same distribution with the same medians versus the alternative hypothesis, i.e., two samples come from different distributions where PCLFS has the higher median. According to the p-values of the tests for each scenario, out of 270 tests, 259 tests rejected the null hypothesis. Details of the test which could not reject the null hypotheses are shown in Table 4.3 and those situations can be identified as the number of informative features in the samples is 1, 2, or 30. Apart from that, we reject the null hypotheses for all the other combinations concluding that the population F1-score median ranks of the PCLFS are greater than the population F1-score median ranks of the Logit-RFE method.

Table 4.3: Wilcoxon signed-rank test results for not rejecting the null hypothesis

sample_size	imbalance_rate	n_informative	p_value	decision
200	[0.5, 0.5]	1	0.946186261	no evidence to reject H0
200	[0.7, 0.3]	1	0.19340269	no evidence to reject H0
200	[0.7, 0.3]	2	0.060100433	no evidence to reject H0
200	[0.9, 0.1]	1	0.71128228	no evidence to reject H0
500	[0.5, 0.5]	1	0.258981528	no evidence to reject H0
500	[0.7, 0.3]	1	0.737989015	no evidence to reject H0
1000	[0.5, 0.5]	1	0.102578085	no evidence to reject H0
1000	[0.5, 0.5]	30	0.335634408	no evidence to reject H0
1000	[0.7, 0.3]	30	0.344183533	no evidence to reject H0
1000	[0.9, 0.1]	1	0.532166148	no evidence to reject H0
1000	[0.9, 0.1]	30	0.266472013	no evidence to reject H0

4.5 Application

4.5.1 SPECTF heart data

To analyze the behavior of models on a real-world data set, we consider the publicly available Single-photon emission computed tomography (SPECT) heart data set (Kurgan et al., 2001; Krzysztof et al., 1997), which describes diagnosing cardiac abnormalities using SPECT. SPECT is an imaging technique where a radioisotope is used to produce 3D images of a patient, using gamma rays (Bruyant, 2002). The data set has classified each of the patients into two categories: normal and abnormal, by considering the diagnosis of images. This data consists of binary class imbalanced data with a higher number of numerical features and a lower number of instances.

The data set consists of 267 SPECT image sets (patients), which were processed to extract features that summarize the original SPECT images. As a result, 44 continuous feature patterns were created for each patient. Hence, it has 267 instances that are described by 45 attributes (44 continuous and 1 binary class). We also divided the data set into two groups, 75% training samples and 25% test samples. The class-imbalanced rate for the data set is 79.4%:20.6%, where the minority class represents the abnormal patients. The imbalance is the same in the training and test set. The data classification with the first two principal components is shown in the score plot in Fig. 4.15. According to the figure, abnormal patients are scattered in a small region whereas the variability of the normal patients is higher than it for the abnormal patients. However, this plot does not give a clear distinction between two categories and it could simply mean that the largest sources of variations are similar in both categories.

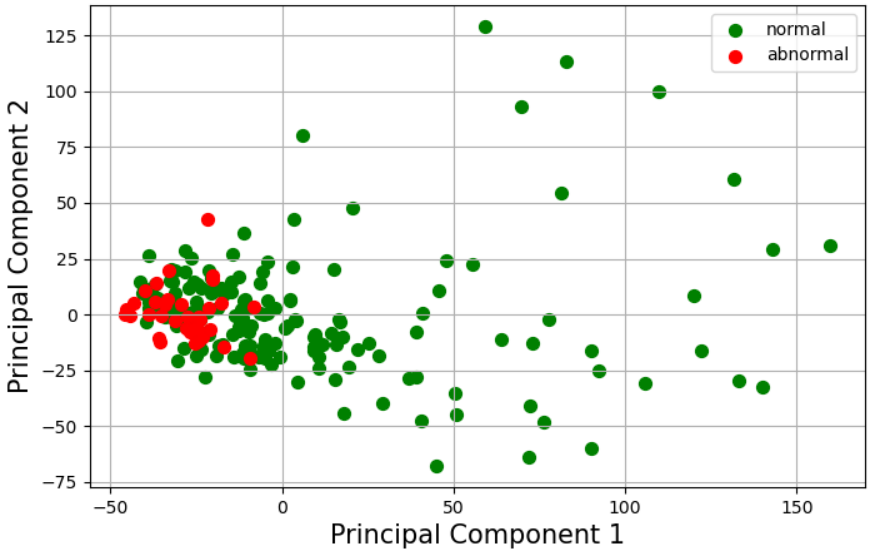

Fig. 4.15. Classification of first two principal components for original data

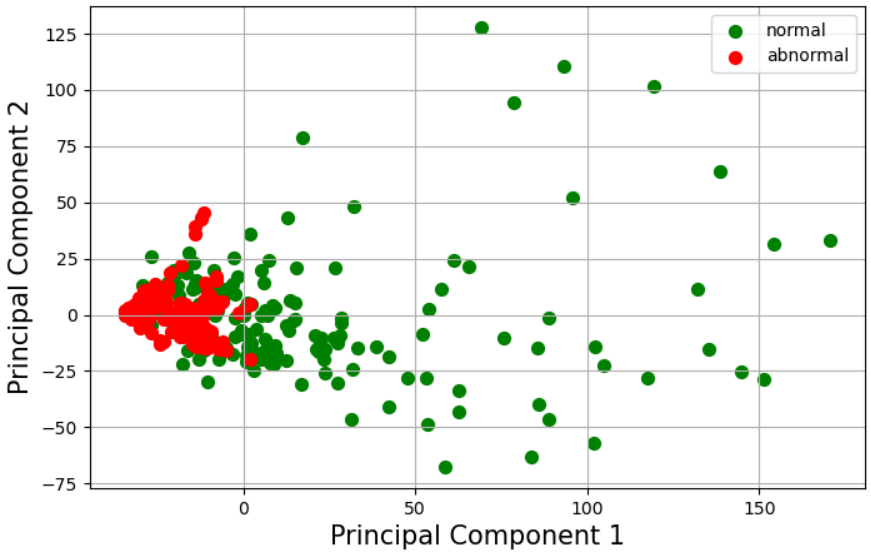

Fig. 4.16. Classification of first two principal components for SMOTE data

Table 4.4: Application results for comparing RFE and PCLFS

SMOTE	Model	RFE				PCLFS			
		#Features	Precision	Recall	F1-score	#Features	Precision	Recall	F1-score
TRUE	Logit	36	0.6154	0.80	0.6957	24	0.6154	0.90	0.6957
	LGBM	27	0.7333	0.55	0.6286	13	0.7857	0.65	0.7027
	Decision Tree	44	0.6250	0.50	0.5556	9	0.7059	0.70	0.6667
	RFC	38	0.6875	0.55	0.6111	42	0.8571	0.70	0.7059
	SVM-Linear	30	0.6522	0.75	0.6977	12	0.7083	0.95	0.7727
FALSE	Logit	30	0.6667	0.40	0.5000	44	0.6923	0.45	0.5455
	LGBM	15	0.6923	0.45	0.5455	15	0.8333	0.50	0.6250
	Decision Tree	27	0.7273	0.40	0.5161	9	1.0000	0.55	0.5946
	RFC	9	0.7143	0.25	0.3704	11	1.0000	0.30	0.4444
	SVM-Linear	21	0.7143	0.50	0.5882	37	0.9000	0.60	0.6316

Then we apply Synthetic Minority Oversampling Technique (SMOTE) to handle imbalanced data to achieve higher accuracy in classification models. The SMOTE aims to balance class distribution by randomly increasing minority class examples by creating similar instances. The classification for SMOTE data with the first two principal components is shown in Fig. 4.16.

4.5.2 Feature selection and result comparison of two methods

We applied the PCLFS and RFE feature selection methods to select features from the sample by fitting them on five classification methods. The final ranked feature list for the RFE was obtained by considering the feature importance of the selected subset and the feature raking of the original classification model. Results are shown in Table 4.4. According to the accuracy measures, F1-score, Precision, and Recall, the PCLFS performs better than the existing RFE method in feature selection.

Feature rankings (with SMOTE) relevant to each method considered are

Table 4.5: Feature ordering ranks for each feature selection method (with SMOTE)

Features	PCLFS	Logit_RFE	Lgbm_RFE	DT_RFE	RFC_RFE	SVM_RFE
F22S	1	28	4	1	1	21
F21S	2	20	33	19	7	20
F21R	3	29	12	19	16	16
F22R	4	13	1	19	6	17
F13S	5	10	9	19	4	8
F15S	6	34	24	12	17	35
F13R	7	23	41	16	12	26
F20S	8	12	2	3	2	1
F15R	9	17	36	19	31	25
F18S	10	39	12	19	30	27
F20R	11	43	12	15	10	36
F5S	12	2	6	19	29	4
F9S	13	18	28	19	34	31
F1S	14	42	12	18	18	28
F5R	15	19	24	17	38	14
F4S	16	38	10	5	22	44
F9R	17	37	18	19	21	42
F4R	18	25	32	19	24	24
F19S	19	36	28	19	20	7
F1R	20	22	18	19	26	29
F18R	21	6	34	19	41	22
F8S	22	21	7	4	5	6
F8R	23	44	36	6	19	12
F12S	24	30	12	19	42	9
F12R	25	15	40	19	40	37
F14S	26	14	4	19	25	11
F19R	27	26	36	13	33	5
F2S	28	40	24	2	13	41
F3S	29	41	28	19	15	23
F10S	30	9	36	7	43	38
F6S	31	33	28	14	36	34
F7R	32	8	18	19	14	19
F3R	33	35	41	19	23	32
F16S	34	1	18	9	3	2
F7S	35	5	12	19	8	3
F14R	36	4	10	19	32	10
F6R	37	24	41	19	39	43
F11S	38	27	7	8	11	18
F17S	39	7	18	11	9	39
F2R	40	32	23	19	27	40
F10R	41	11	34	19	28	33
F16R	42	3	24	10	37	15
F11R	43	31	44	19	44	30
F17R	44	16	3	19	35	13

shown in Table 4.5. Then the Bland-Altman (B&A) plots (Bland and Altman, 1999) were used to determine the agreement between two pairs of quantitative rankings. The method quantifies agreement between two quantitative measurements by constructing limits of agreement for each RFE model with PCLFS. These statistical limits were calculated using the mean and the standard deviation (s) of the differences between the two ranks. First, we checked the assumptions of normality of differences using the Shapiro normality test, and all differences were concluded as normally distributed. Since the points on the plots (Fig. 4.17) are scattered above and below zero and within limits, it suggests no consistent bias of one approach versus the other. The straight line on the decision tree-RFE plot and the results from Table 4.4 indicate that the decision tree-RFE has given equal ranking for several features while PCLFS ranks them differently.

Finally, we selected the relevant feature subset using PCLFS and RFE for each classification model, and the features selected by each method are presented in table 4.6.

In summary, the simulation study has proven the ability of the PC loadings to order features according to the importance compared to the other method considered and the approach of selecting important feature subsets using the PCLFS method, which performs better than the existing RFE. The application results ultimately ensure the accuracy of the findings.

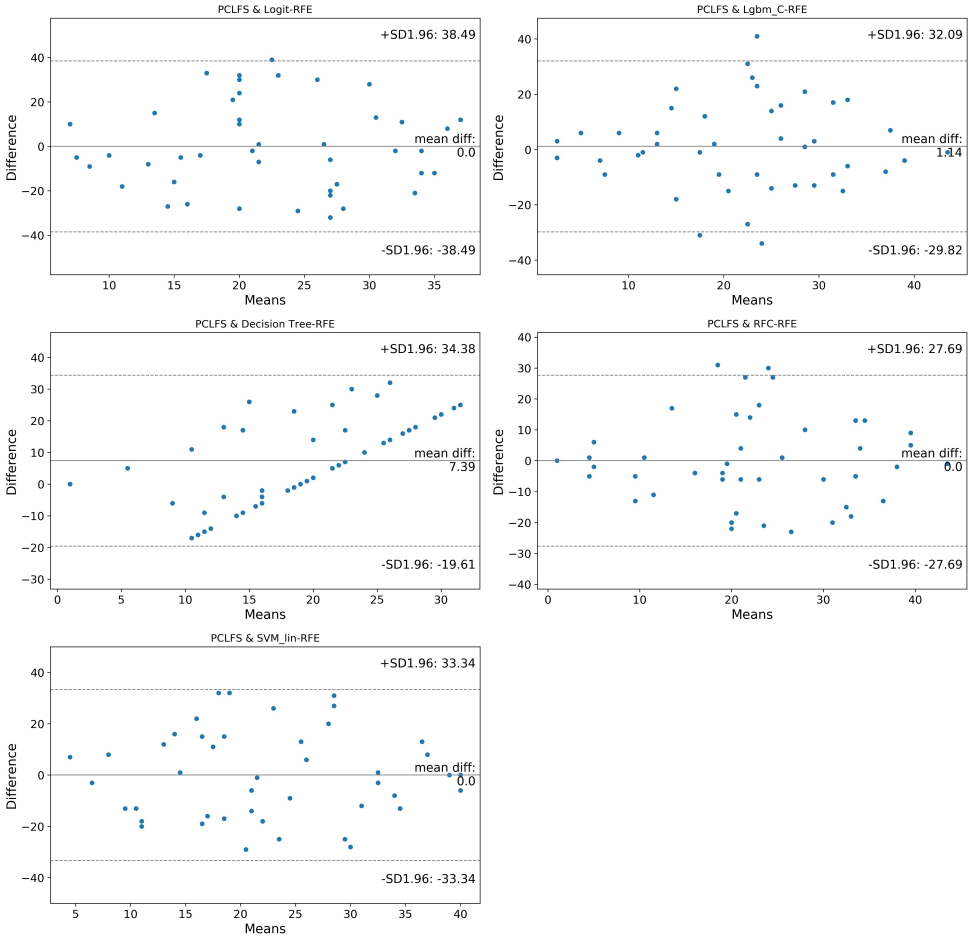

Fig. 4.17. Bland and Altman plot for data from Table 4.5 by comparing PCLFS with each RFE model, with the representation of the limits of agreement (dotted line), from -1.96s to +1.96s.

Table 4.6: Feature Selection with SMOTE. Red check marks will be discussed in Section 5.3.

		PCLFS					RFE				
feature	**pclfs_order**	**logit**	**lgbm_c**	**dt**	**rfc**	**svm_lin**	**logit**	**lgbm_c**	**dt**	**rfc**	**svm_lin**
F22S	1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F21S	2	✓	✓	✓	✓	✓	✓		✓	✓	✓
F21R	3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F22R	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F13S	5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
F15S	6	✓	✓	✓	✓	✓	✓		✓	✓	
F13R	7	✓	✓	✓	✓	✓	✓		✓	✓	✓
F20S	8	✓	✓	✓	✓	✓	✓	✓	✓	✓	
F15R	9	✓	✓	✓	✓	✓	✓		✓	✓	✓
F18S	10	✓	✓		✓	✓		✓	✓	✓	✓
F20R	11	✓	✓		✓	✓			✓	✓	
F5S	12	✓	✓		✓	✓	✓	✓	✓	✓	✓
F9S	13	✓	✓		✓		✓		✓		✓
F1S	14	✓			✓			✓	✓	✓	✓
F5R	15	✓			✓		✓		✓	✓	✓
F4S	16	✓			✓			✓	✓	✓	
F9R	17	✓			✓				✓	✓	✓
F4R	18	✓			✓		✓	✓	✓	✓	✓
F19S	19	✓			✓			✓	✓	✓	
F1R	20	✓			✓		✓	✓	✓	✓	✓
F18R	21	✓			✓		✓		✓		✓
F8S	22	✓			✓		✓	✓	✓	✓	✓
F8R	23	✓			✓			✓	✓	✓	✓
F12S	24	✓			✓		✓	✓	✓		✓
F12R	25				✓		✓		✓		
F14S	26				✓			✓	✓	✓	✓
F19R	27				✓		✓	✓	✓	✓	✓
F2S	28				✓				✓	✓	
F3S	29				✓		✓	✓	✓	✓	✓
F10S	30				✓		✓		✓		
F6S	31				✓		✓	✓	✓	✓	✓
F7R	32				✓		✓	✓	✓	✓	✓
F3R	33				✓		✓		✓	✓	
F16S	34				✓		✓	✓	✓	✓	✓
F7S	35				✓		✓	✓	✓	✓	✓
F14R	36				✓		✓	✓	✓		✓
F6R	37				✓		✓		✓	✓	
F11S	38				✓		✓		✓	✓	✓
F17S	39				✓		✓	✓	✓	✓	
F2R	40				✓		✓		✓	✓	✓
F10R	41				✓		✓		✓	✓	✓
F16R	42				✓		✓	✓	✓		✓
F11R	43						✓		✓		✓
F17R	44						✓	✓	✓	✓	✓

Chapter 5

Combining Proposed Methods

5.1 Introduction

The main objective of the thesis is to introduce a new feature selection method that selects the minimal number of features, including the most informative features. Hence, we combine both suggested methods in Chapter 3 and Chapter 4 to develop a solidified answer for the problems facing in feature selection. The PCLFS method is extended to achieve the minimum number of informative features with higher accuracy or a minimum loss of accuracy.

For the PCLFS-extended method, grid scores ($\mathbf{g}$) are the F1-scores such that g_i corresponds to the F1-score of the i^{th} feature subset with the first i features of the PCLFS ordered feature list. The rest of the notations are the same as for the RFE discussed in Section 3.3.3.

5.2 Simulation Results

This section illustrates the results obtained through synthetic samples and the simulation study results on all three methods, existing RFE, proposed PCLFS, and PCLFS-extended. Fig. 5.1 shows results of a selected simulated data set, which contained 12 informative features out of 30 features, 1000 sample size, and 70%:30% imbalance rate. Here, the maximum tolerable F1-score is taken as 0.03. Hence, the threshold per feature is 0.001.

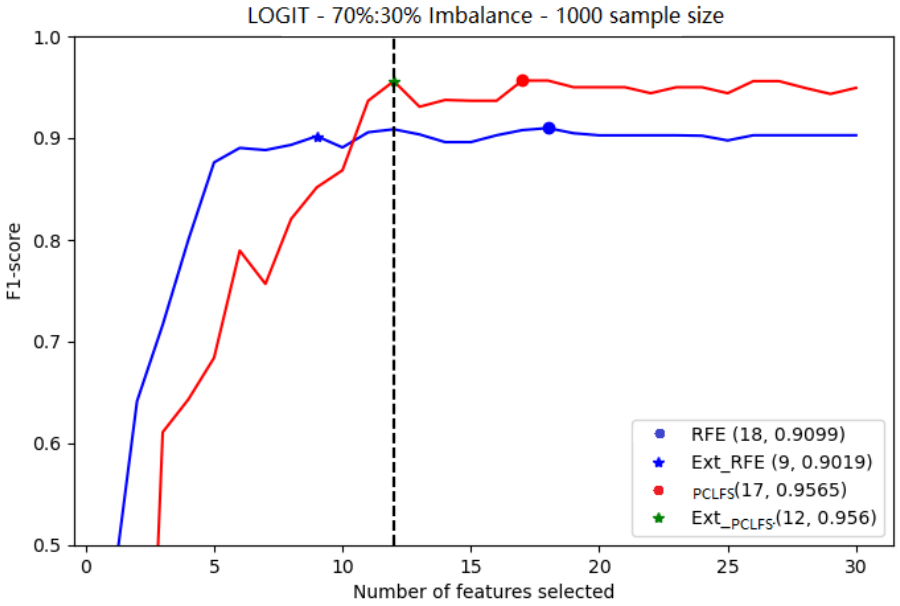

Fig. 5.1. Comparison between RFE CV F1-scores and PCLFS F1-scores for each feature subset.

According to Fig. 5.1, initially, the original RFE gives 18 features with an F1-score of 0.9099, and it reduces to 9 features by the first proposed method with only 0.008 reductions of the F1-score. In the meantime, the PCLFS

method chooses only 17 features, with a 0.0466 increment of the F1-score compared to the original RFE. Combining with the extended method, it selects 12 features (the number of features supposed to be selected) with 0.956 of the F1-score.

To capture the variability of the final F1-scores of each method, we then conducted a simulation study to determine the necessity of the suggested combined approach. One hundred samples are simulated from each scenario to reduce the variability in experimental results, while the number of informative features is increased from 1 to the total number of features. All features are classified as informative or non-informative. No redundant features or repeated features are included in simulated data sets. We generated data for 50%:50% balanced data and two other imbalance rates, 70%:30% and 90%:10%. Two sample sizes with 200 and 1000 sample units were also examined. Most importantly, in this analysis, the models were fitted on original data and re-sampled data with SMOTE. Here, the results are only illustrated for the logistic regression model. Appendix C contains results for other classification models, with highly imbalanced data with 90%:10% rate and a sample size of 1000.

5.2.1 Simulation results without SMOTE

Similar results were obtained for RFE and PCLFS comparison as in Section 4.4.2 and are shown in Fig. 5.2 and 5.4 for two different sample sizes 200 and 1000 for the Logit classifier. However, we newly compare the extended version of PCLFS (PCLFS-ext), which gives a higher feature selection correct percentage for an insignificantly smaller F1-score over the other two methods.

To further understand the selection of features, we plotted the number of selected features and feature selection true positive rate (TPR_{fs}) against the number of informative features given. Feature selection TPR was calculated using the equation explained in Appendix A.1. For the original data, PCLFS and PCLFS-extended methods pick a relatively larger number of features than RFE. Nevertheless, the feature selection TPR is significantly higher in the proposed methods. These results are shown in Fig. 5.3 and 5.5. We note that when the sample size is smaller, the PCLFS-extended method is not tempted to pick a lower number of features in highly imbalanced data under the given threshold of 0.0017.

5.2.2 Simulation results with SMOTE

We repeated the same procedure for SMOTE data with the Logit classifier and results are shown in Fig. 5.6, 5.7, 5.8 and 5.9. Except for having lower feature selection correct percentages for highly imbalanced data with smaller sample sizes (bottom right graph in Fig. 5.6), in all the other scenarios, PCLFS extended version performs much better. When the number of informative features in the data set is greater than four, the PCLFS extended version has a higher F1-score, higher feature selection correct percentages, and higher feature selection TPR. For the highly imbalanced data with a larger sample size, PCLFS and PCLFS-extended methods even pick a lower number of features than RFE when there are few informative features in the data set (Fig. 5.9).

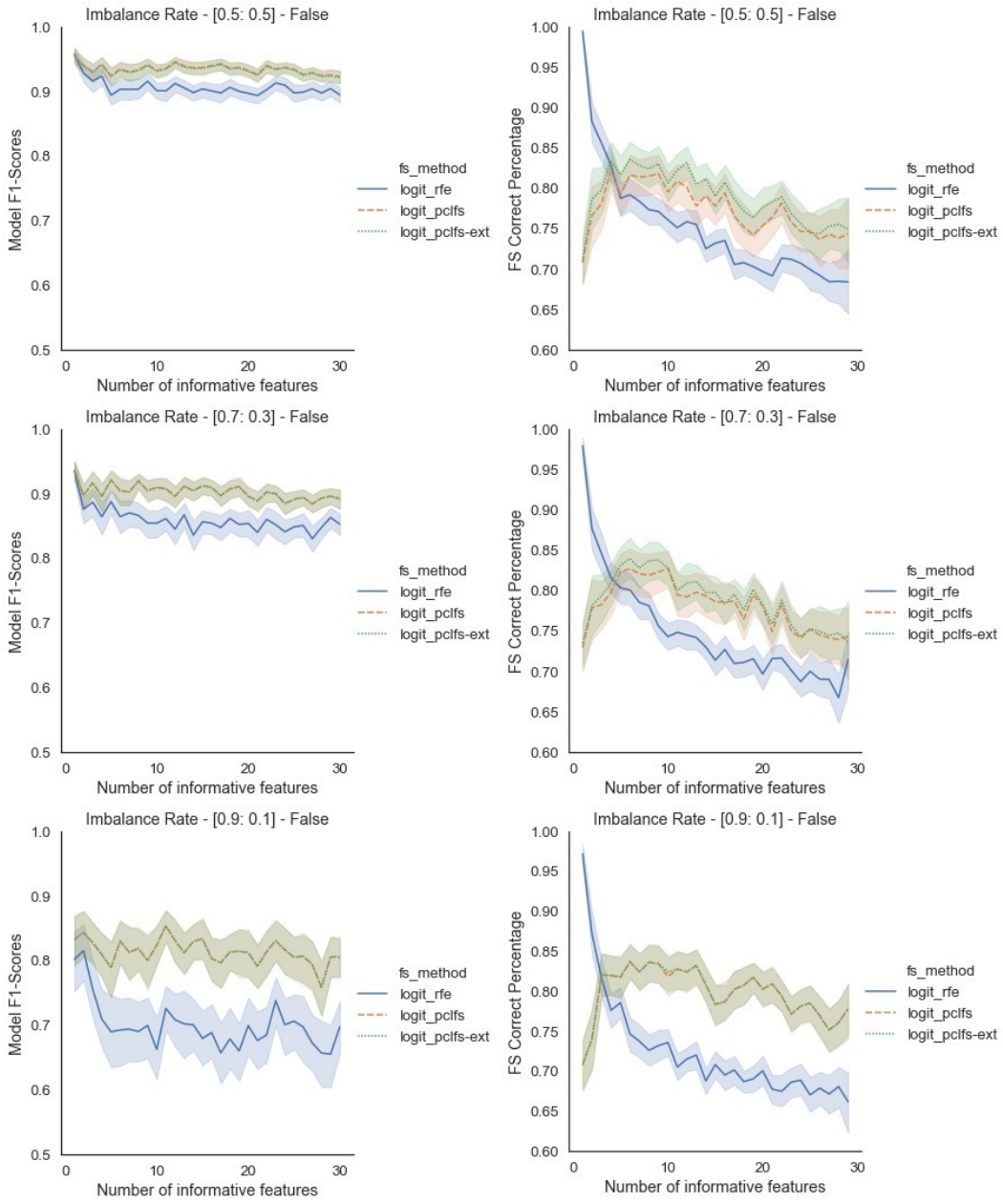

Fig. 5.2. Final model F1-scores and feature selection correct percentages for the Logit model, without SMOTE when sample size is 200 and threshold is 0.0017.

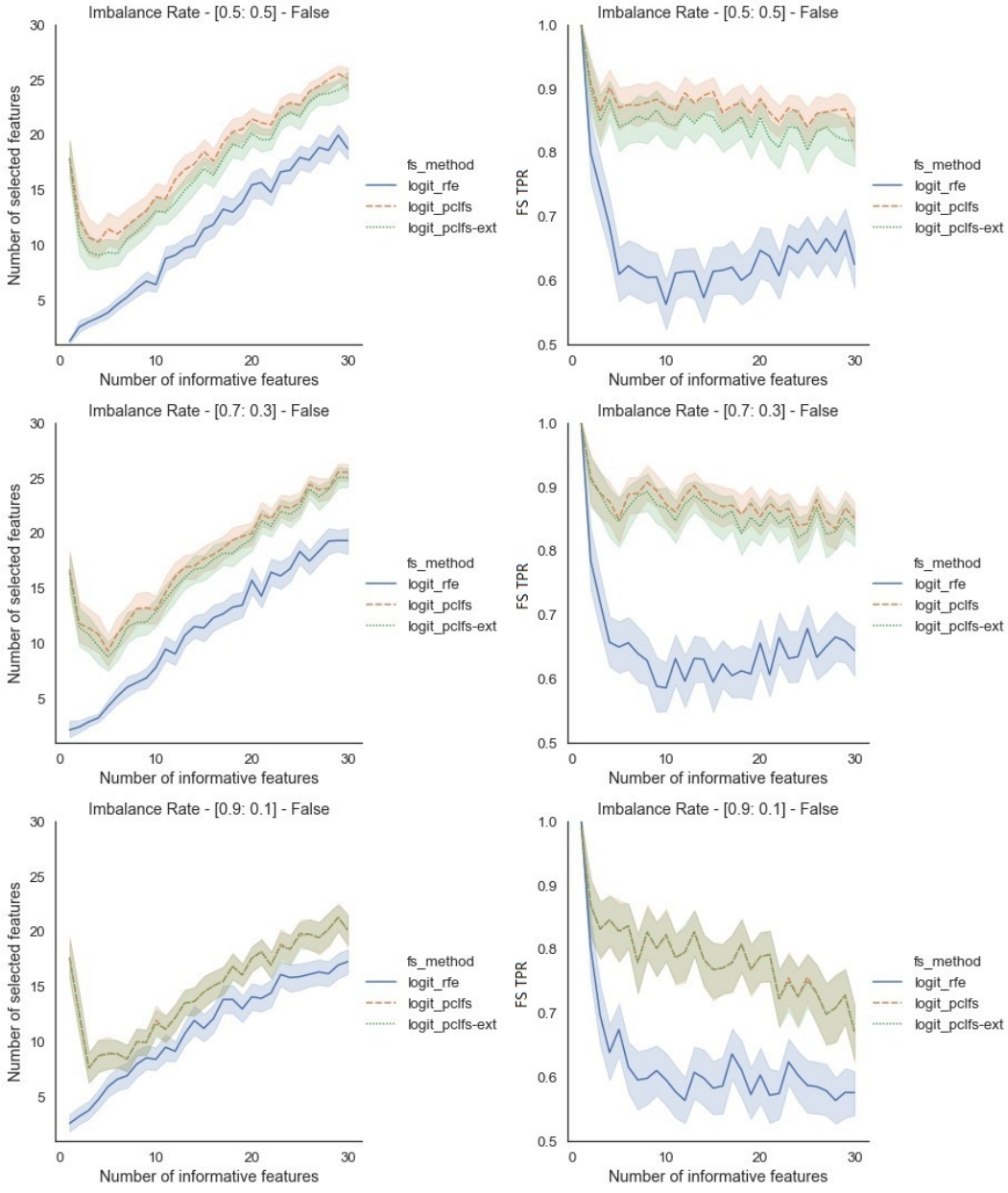

Fig. 5.3. Selected number of features and feature selection TPR for the Logit model, without SMOTE when sample size is 200 and threshold is 0.0017.

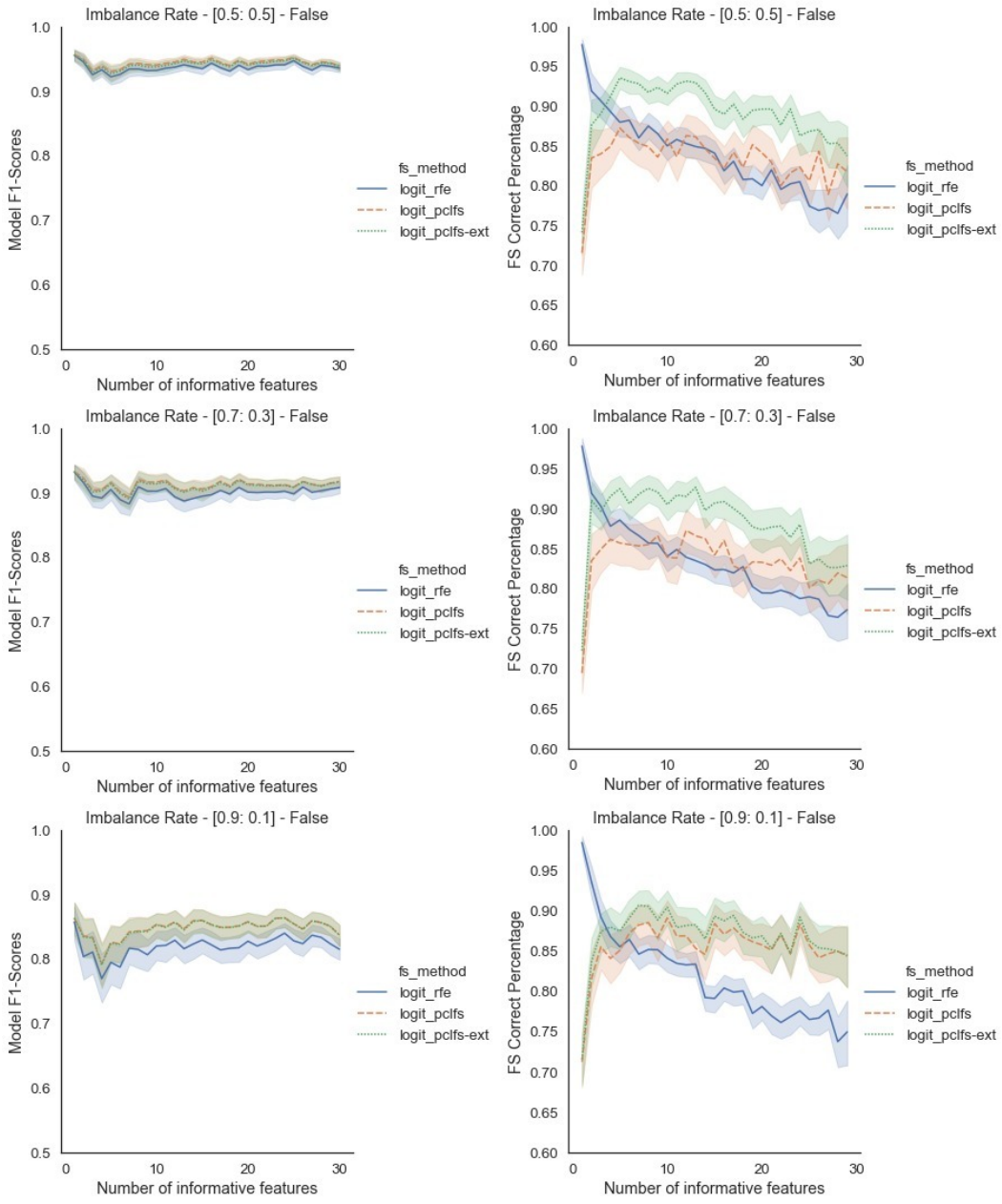

Fig. 5.4. Final model F1-scores and feature selection correct percentages for the Logit model, without SMOTE when sample size is 1000 and threshold is 0.0017.

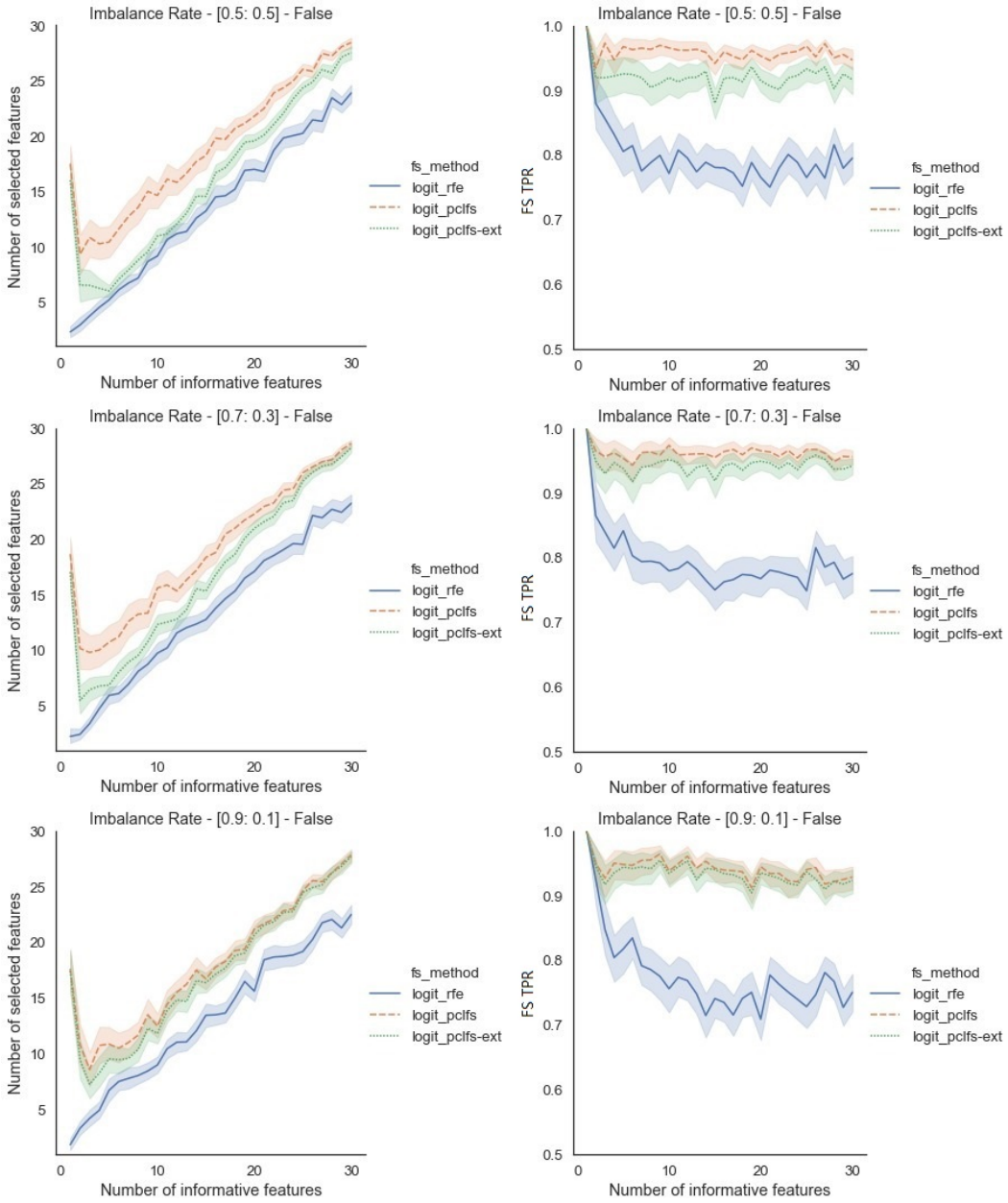

Fig. 5.5. Selected number of features and feature selection TPR for the Logit model, without SMOTE when sample size is 1000 and threshold is 0.0017.

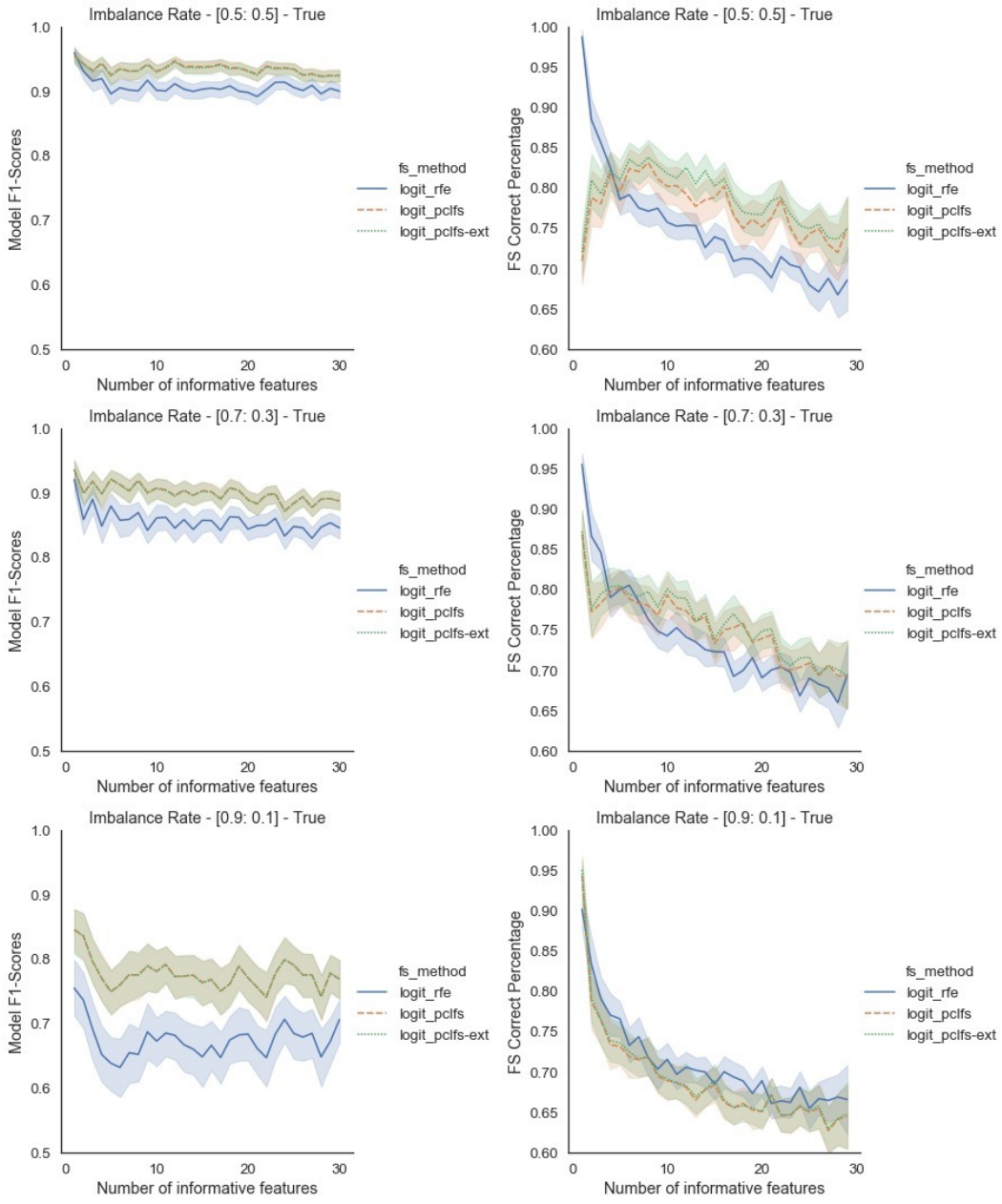

Fig. 5.6. Final model F1-scores and feature selection correct percentages for the Logit model, with SMOTE when sample size is 200 and threshold is 0.0017.

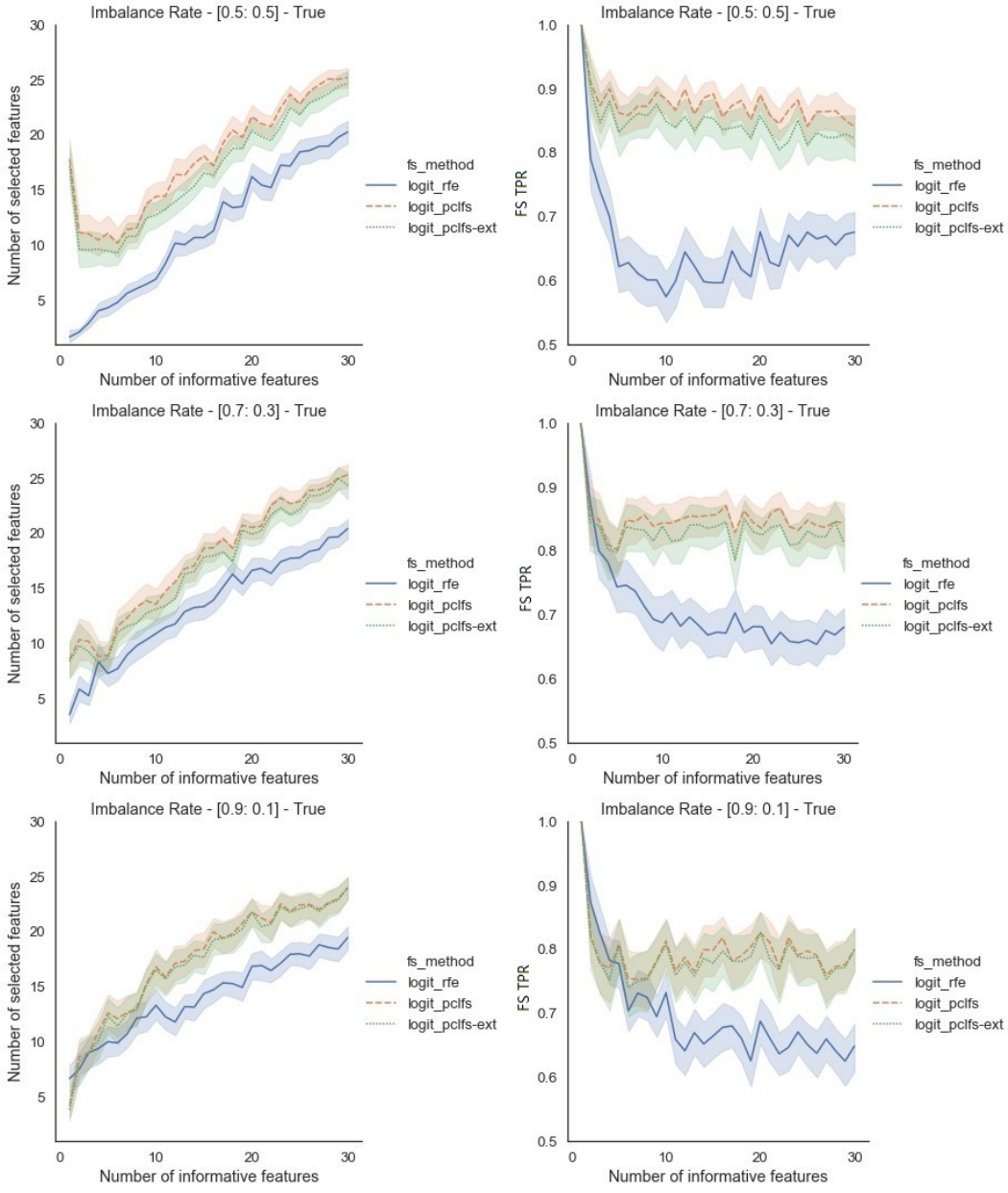

Fig. 5.7. Selected number of features and percentage of informative features selected for the Logit model, with SMOTE when sample size is 200 and threshold is 0.0017.

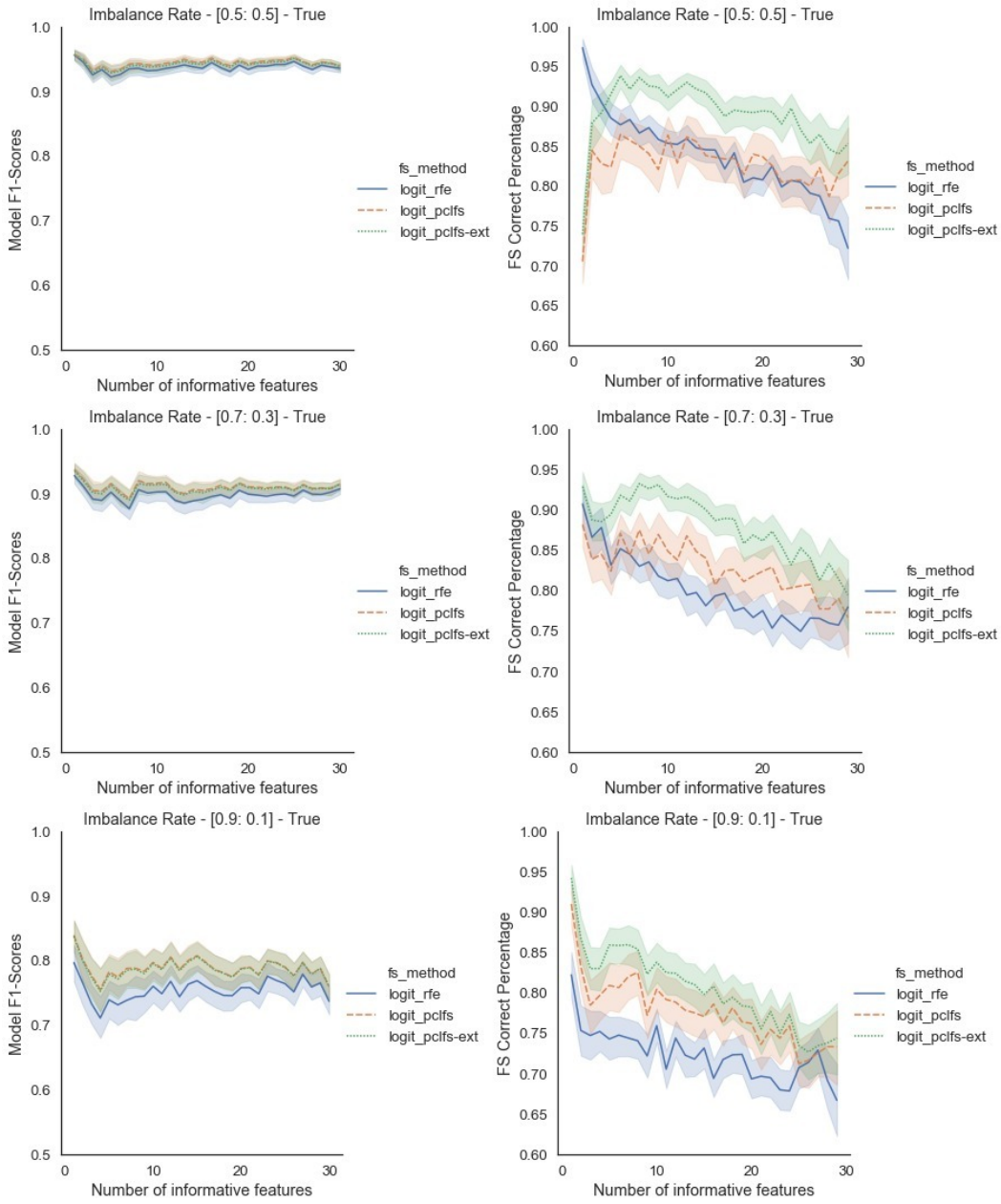

Fig. 5.8. Final model F1-scores and feature selection correct percentages for the Logit model, with SMOTE when sample size is 1000 and threshold is 0.0017.

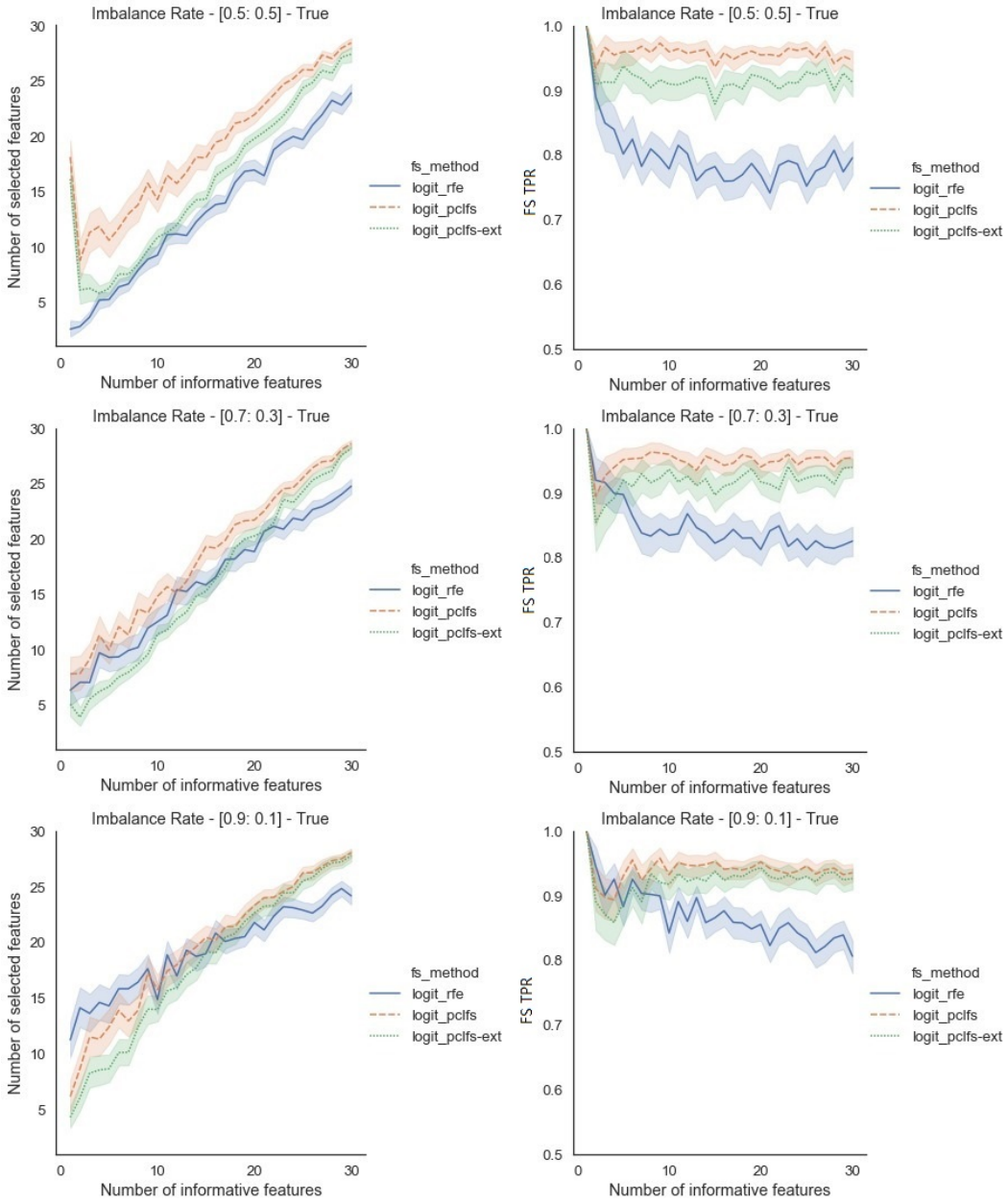

Fig. 5.9. Selected number of features and percentage of informative features selected for the Logit model, with SMOTE when sample size is 1000 and threshold is 0.0017.

5.3 Application Results

To analyze the behavior of models on a real-world data set, the same data set used in Chapter 4, the publicly available Single-photon emission computed tomography (SPECT) heart data set (Kurgan et al., 2001; Krzysztof et al., 1997), and compare the proposed PCLFS and PCLFS-extended model results with the final F1-scores of the existing RFE method. The results are shown in Table 5.1. The PCLFS-Extended selected feature subset with SMOTE is shown in red checkmarks in Table 4.6. For SMOTE data, PCLFS selects a smaller number of features than RFE with a higher F1-score for all the classification models. It further reduces the number of features considerably in the PCLFS-extended method for the Logit, decision tree, and RFC models, and the last two columns of the Table 5.1 depicts the reduction/increment of the percentages of features and the F1-scores over RFE and the proposed method where,

$$\text{Feature reduction/(increment)}\% = \frac{\text{Number of features reduced/(increased)}}{\text{Total number of features}}.$$

Fig. 5.10 display how the PCLFS-extended version picks a lesser number of features with similar performance with a maximum tolerable F1-score of 0.05, hence the threshold of 0.0011.

Table 5.1: Final F1-score comparison between RFE and proposed methods (PCLFS/PCLFS-Extended (t=0.0011)).

SMOTE	Method	Basic		**RFE**		PCLFS		**PCLFS-Extended**		Feature reduction%/ (increment%)	F1-score (reduction)/ increment
		#Features	F1-scores	#Features	F1-scores	#Features	F1-scores	#Features	F1-scores		
TRUE	Logit	44	0.6809	36	0.6957	24	0.6957	11	0.6939	56.8%	(0.0018)
	LGBM	44	0.6667	27	0.6286	13	0.7027	-	-	31.8%	0.0741
	Decision Tree	44	0.5556	44	0.5556	9	0.6667	3	0.6666	93.2%	0.1110
	RFC	44	0.6486	38	0.6111	42	0.7059	12	0.6842	59.0%	0.0731
	SVM-Linear	44	0.6511	30	0.6977	12	0.7727	-	-	40.9%	0.0750
FALSE	Logit	44	0.5455	30	0.5000	44	0.5455	-	-	(31.8%)	0.0455
	LGBM	44	0.6250	15	0.5455	15	0.6250	-	-	0.0%	0.0795
	Decision Tree	44	0.5294	27	0.5161	9	0.5946	-	-	40.9%	0.0785
	RFC	44	0.2609	9	0.3704	11	0.4444	-	-	(4.5%)	0.0740
	SVM-Linear	44	0.5946	21	0.5882	37	0.6316	-	-	(36.4%)	0.0434

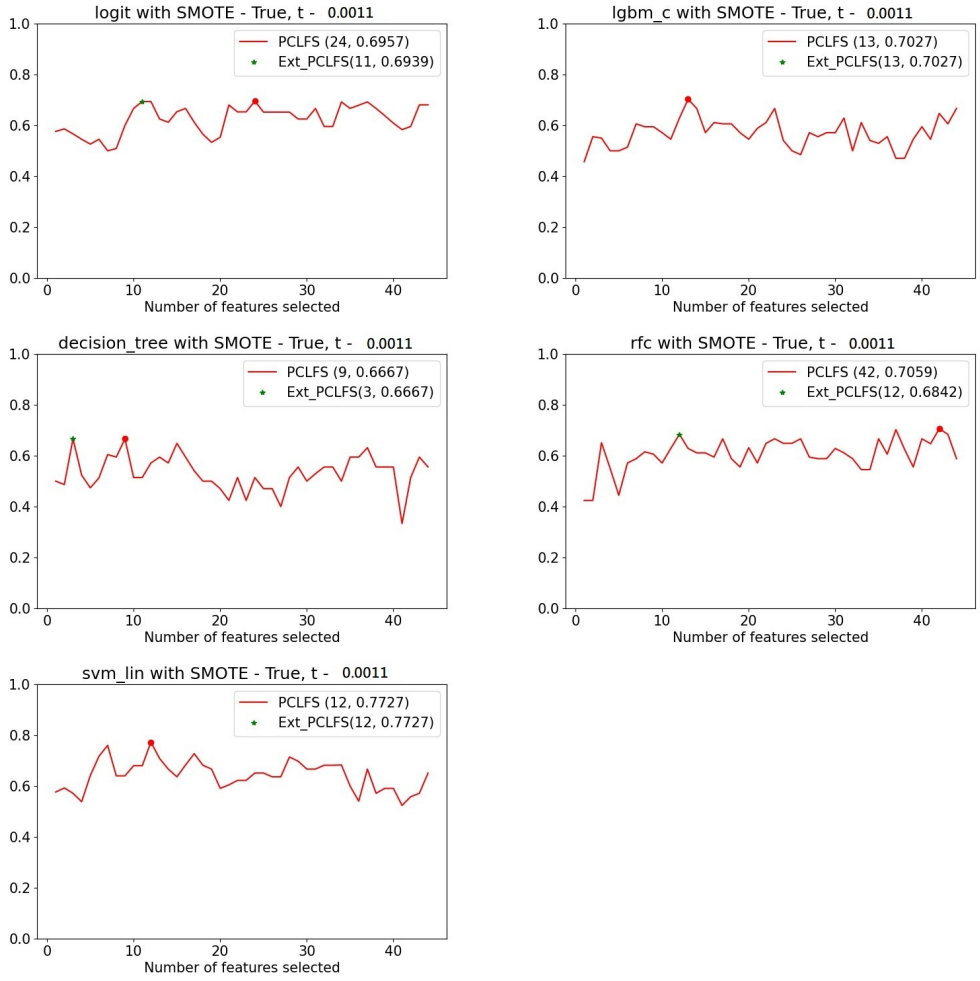

Fig. 5.10. Selecting smaller number of features under the threshold of 0.0011. Red point indicates the PCLFS selection whereas the green star indicated the extended PCLFS method selection.

Chapter 6

Discussion

Selecting a subset from the original feature set is called feature selection, and it has become an important aspect of matured machine learning methods. Feature selection is also known as variable selection, feature reduction, attribute selection, or variable subset selection (Liu and Yu, 2005). This process is essential in practice for many reasons, especially if we have to collect data from costly sources such as sensors, patients, blood samples, etc. In such situations, we have to limit the number of features to a reasonable value; still, identifying the most important feature subset is crucial. Not only that, but having a smaller number of features also increases the computational efficiency and the prediction performances of the model. As a solution, we have proposed a new approach for the existing wrapper methods to select a minimal number of important features with similar performance. Hence, this is an important contribution as it reduces the cost, especially in data collection.

Most of the wrapper feature selection methods compare scores of several feature subsets and select the one that gives the maximum score. There are other selections of a lower number of features with lower-score, yet with little

difference in score. This thesis proposes and applies an extended version of selecting a subset of features with minimal features instead of having the subset with the maximum score. Recursive feature elimination is one of the most commonly used wrapper methods. In standard RFE, a feature is eliminated if it is the least important to predicting, and features are ranked according to the model's strength by considering the performance scoring method. It obtains the best feature subset by comparing the scores, where the feature subset which gives the best score is identified as the optimal feature subset. Still, some other feature subsets practically reduce the number of features with a minimal loss of score.

Our first proposed method assesses the number of features below the maximum and calculates the most beneficial smallest number of features and the feature subset with a tolerable score deduction. We compare the proposed approach to RFE results on simulated data sets and a real-world data set. It is clearly shown that the proposed method makes a reasonable improvement over RFE results.

The threshold plays a vital role in the introduced algorithm as the numerator, the maximum tolerable F1-score, is decided using the user's domain knowledge and desire. The selection of the threshold is sensitive to the imbalance rate of the data. For highly imbalanced data, we can use a relatively larger threshold to achieve a similar result. Although we have considered only five classification models in examples, like in RFE, the proposed method can also be fitted on any classification model that has an inherent quantification of the importance of a feature such as RFE on non-linear kernels with SVM (Xue et al., 2018).

Most wrapper feature selection methods use an already ordered feature list as an initial step of selecting features. Still, the question is, how strong

and reliable is the ordering ability of the method used to rank the features according to their importance. Also, the ordering feature set would highly depend on the classification model used in the problem. In this research, we compare four different feature ordering techniques. Using synthetic data, we identify the best feature ordering mechanism as a solution to this issue.

The absolute sum of principal component loadings orders the features more informatively than other selected methods when the number of informative features is more than four in the sample. Therefore, we then introduced a feature selection method (PCLFS), which uses the absolute sum of principal component loadings to rank the features and a sequential search method to select the feature subset. The PCLFS performs much better with smaller sample sizes and highly imbalanced data sets. In most wrapper feature selection methods, the search space for d features is 2^d, which is a known issue when there are many features. But, our approach eliminates this issue by reducing the search space to d. Further, the suggested method identifies the most informative features first. Hence, without applying sequential feature selection, the user can also have any affordable number of features by only considering the PC loading order.

The user can select the number of PCs by considering the contribution of each principal component to the total explained variance. The validity of the assumptions and limitations of the PCA ((i). Linearity, (ii). Large variances have important structure, and (iii). The principal components are orthogonal) are important for the suggested approach (Shlens, 2014). Also, the multiple variables need to be measured at the continuous level and need to have a large enough sample size. Data also must be suitable for data reduction, and there should be no significant outliers in the data set. Although principal components

try to cover maximum variance among the features in a data set, if we do not select the number of Principal Components with care, it may miss some information compared to the original list of features. Hence, the impact of the number of principal components for the suggested method also should be examined in the future.

Finally, we merged two proposed methods, PCLFS and extended version of wrapper feature selection, and concluded with a smaller feature subset that picks the most informative features while giving higher F1-score accuracy than the existing method irrespective of re-balancing the data set.

Since we are using the classification models in the latter stage of the PCLFS method, ordering features are entirely independent of the classification model. The finding and facts of the thesis are subject to the considerations in synthetic data generation. The redundant and repeated features were not considered for the synthetic data generation and assumed that the data set consisted of informative and non-informative features only. However, in real applications, the underlying truth is unknown. Hence, the impact of adding redundant and repeated features can also be analyzed in future research.

In this research, synthetic simulations, computations, and related experiments were done using python, and the WestGrid facility was used due to the computational intensity.

Bibliography

Ali, S. I., H. S. M. Bilal, M. Hussain, J. Hussain, F. A. Satti, M. Hussain, G. H. Park, T. Chung, and S. Lee (2020). Ensemble feature ranking for cost-based non-overlapping groups: A case study of chronic kidney disease diagnosis in developing countries. *IEEE Access 8*, 215623–215648. (Cited on page 2.)

Bellman, R. (1957). *Dynamic programming.* Princeton: Princeton University Press. (Cited on pages 1 and 55.)

Bland, J. M. and D. G. Altman (1999). Measuring agreement in method comparison studies. *8*(2), 135–60. (Cited on page 86.)

Breiman, L. (2001). Random forests. *Machine learning 45*(1), 5–32. (Cited on pages 14, 29 and 56.)

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and regression trees.* Monterey, CA: JWadsworth Brooks/Cole Advanced Books Software. (Cited on pages 14, 29, 56 and 60.)

Bruyant, P. P. (2002). Analytic and iterative reconstruction algorithms in spect. *Journal of Nuclear Medicine 43*(10), 1343–1358. (Cited on page 82.)

Butcher, B. and B. J. Smith (2020). Feature engineering and selection: A practical approach for predictive models: by max kuhn and kjell johnson.

boca raton, fl: Chapman hall/crc press, 2019, xv + 297 pp., isbn: 978-1-13-807922-9. *The American statistician 74*(3), 308–309. (Cited on page 59.)

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). Smote: Synthetic minority over-sampling technique. *The Journal of artificial intelligence research 16*, 321–357. (Cited on pages 4, 11, 48 and 57.)

Chawla, N. V., N. Japkowicz, and A. Kołcz (2004). Editorial: Special issue on learning from imbalanced data sets. *SIGKDD Explorations 6*, 1–6. (Cited on page 11.)

Chen, X. and J. C. Jeong (2007). Enhanced recursive feature elimination. In *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pp. 429–435. (Cited on page 31.)

Chen, X. and M. Wasikowski (2008). Fast: A roc-based feature selection metric for small samples and imbalanced data classification problems. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, New York, NY, USA, pp. 124–132. Association for Computing Machinery. (Cited on page 31.)

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning 20*(3), 273–297. (Cited on pages 29 and 57.)

Dunteman, G. (1989). Using principal components to select a subset of variables. In *Principal Components Analysis*, Quantitative Applications in the Social Sciences. Newbury Park: SAGE Publications, Inc. (Cited on page 58.)

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics 29*(5), 1189–1232. (Cited on pages 14, 29 and 57.)

Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering 2*, 42–47. (Cited on page 11.)

Guo, Q., W. Wu, D. L. Massart, C. Boucon, and S. De Jong (2002). Feature selection in principal component analysis of analytical data. *Chemometrics and Intelligent Laboratory Systems 61*(1-2), 123–132. (Cited on pages 2, 14, 29 and 56.)

Guyon, I., A. Elisseeff, and L. P. Kaelbling (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research 3*(7-8), 42–47. (Cited on page 61.)

Guyon, I., J. Weston, S. Barnhill, and V. Vapnik (2002). Gene selection for cancer classification using support vector machines. *Machine learning 46*(1), 389–422. (Cited on pages 8, 9, 29 and 56.)

Hadden, J., A. Tiwari, R. Roy, and D. Ruta (2007). Computer assisted customer churn management: State-of-the-art and future trends. *Computers Operations Research 34*(10), 2902–2917. (Cited on page 47.)

Hastie, T., R. Tibshirani, and J. H. Friedman (2009). *The elements of statistical learning : data mining, inference, and prediction* (2nd ed. ed.). Springer series in statistics. New York: Springer. (Cited on pages 15, 29 and 56.)

Hotelling, H. (1933). *Analysis of a complex of statistical variables into principal components*. Baltimore: Warwick York. (Cited on pages 26 and 61.)

Hua, J., Z. Xiong, J. Lowey, E. Suh, and E. Dougherty (2005). Optimal number of features as a function of sample size for various classification rules. *Bioinformatics (Oxford, England) 21*, 1509–15. (Cited on page 9.)

Huang, B., M. T. Kechadi, and B. Buckley (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications 39*(1), 1414 – 1425. (Cited on page 47.)

IBM (2019). Telco customer churn. [Online; accessed 11-July-2019]. (Cited on page 47.)

Kohavi, R. and G. H. John (1997). Wrappers for feature subset selection. *Artificial intelligence 97*(1-2), 273–324. (Cited on pages 9, 29 and 56.)

Kotsiantis, S., D. Kanellopoulos, and P. Pintelas (2005). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering 30*, 25–36. (Cited on page 10.)

Krzysztof, J. C., K. W. Daniel, and L. Ning (1997). Clip3: Cover learning using integer programming. *26*(5). (Cited on pages 82 and 101.)

Kuhn, M. (2013). *Applied Predictive Modeling*. New York, NY. (Cited on pages 1 and 55.)

Kumari, B. and T. Swarnkar (2011, 01). Filter versus wrapper feature subset selection in large dimensionality micro array: A review. *International Journal of Computer Science and Information Technologies 2*, 1048–1053. (Cited on page 9.)

Kurgan, L. A., K. J. Cios, R. Tadeusiewicz, M. Ogiela, and L. S. Goodenday (2001). Knowledge discovery approach to automated cardiac spect diagnosis. *Artificial Intelligence in Medicine 23*(2), 149–169. (Cited on pages 82 and 101.)

Lal, T. N., O. Chapelle, J. Weston, and A. Elisseeff (2006). Embedded methods. In *Feature Extraction*, Studies in Fuzziness and Soft Computing, pp. 137–165. Berlin, Heidelberg: Springer Berlin Heidelberg. (Cited on page 7.)

Lev, J. (1949). The point biserial coefficient of correlation. *Ann. Math. Statist. 20*(1), 125–126. (Cited on page 60.)

Liu, H. and L. Yu (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE transactions on knowledge and data engineering 17*(4), 491–502. (Cited on page 104.)

Mazurowski, M. A., P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassi (2008). Training neural network classifiers for medical decision making: the effects of imbalanced datasets on classification performance. *Neural networks : the official journal of the International Neural Network Society 21*(2-3), 427—436. (Cited on page 10.)

McCullagh, P. and J. A. Nelder (1989). *Generalized linear models* (2nd ed. – ed.). Monographs on statistics and applied probability ; 37. London: Chapman Hall. (Cited on pages 15, 29 and 56.)

Miche, Y., P. Bas, A. Lendasse, C. Jutten, and O. Simula (2007a). Advantages of using feature selection techniques on steganalysis schemes. In F. Sandoval, A. Prieto, J. Cabestany, and M. Graña (Eds.), *Computational and Ambient Intelligence*, Berlin, Heidelberg, pp. 606–613. Springer Berlin Heidelberg. (Cited on page 1.)

Miche, Y., P. Bas, A. Lendasse, C. Jutten, and O. Simula (2007b). Advantages of using feature selection techniques on steganalysis schemes. In F. Sandoval, A. Prieto, J. Cabestany, and M. Graña (Eds.), *Computational and Ambient*

Intelligence, Berlin, Heidelberg, pp. 606–613. Springer Berlin Heidelberg. (Cited on pages 7 and 55.)

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill, Maidenhead, U.K. (Cited on page 25.)

Nisbet, R. (2009). *Handbook of statistical analysis and data mining applications*. Amsterdam: Academic Press/Elsevier. (Cited on page 8.)

Nisbet, R. (2012). *Practical text mining and statistical analysis for non-structured text data applications* (1st ed. ed.). Amsterdam: Academic Press. (Cited on page 1.)

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011a). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830. (Cited on pages 4, 22, 23 and 30.)

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011b). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830. (Cited on page 57.)

Peng, Y., Z. Wu, and J. Jiang (2010). A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics 43*(1), 15–23. (Cited on page 61.)

Powers, D. (2008). Evaluation: From precision, recall and f-factor to roc,

informedness, markedness correlation. *Mach. Learn. Technol. 2*. (Cited on page 24.)

Provost, F. and T. Fawcett (2001). Robust classification for imprecise environments. *Machine Learning 42*(3), 203–231. (Cited on page 24.)

Pudil, P., J. Novovičová, and J. Kittler (1994). Floating search methods in feature selection. *Pattern Recognition Letters 15*(11), 1119–1125. (Cited on page 61.)

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. (Cited on page 17.)

Refaeilzadeh, P., L. Tang, and H. Liu (2009). Cross-validation. *Encyclopedia of Database Systems 532–538*, 532–538. (Cited on page 25.)

Ryzin, J. V. (1986). Breiman, leo, friedman, jerome h., olshen, richard a., and stone, charles j., "classification and regression trees" (book review). *Journal of the American Statistical Association 81*(393), 253–. (Cited on page 23.)

Saeys, Y., I. Inza, and P. Larrañaga (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics 23*(19), 2507–2517. (Cited on page 8.)

Samb, M. L., F. Camara, S. Ndiaye, Y. Slimani, and M. A. Esseghir (2012). A novel rfe-svm-based feature selection approach for classification. *International Journal of Advanced Science and Technology 43*. (Cited on pages 1 and 31.)

Shlens, J. (2014). A tutorial on principal component analysis. (Cited on page 106.)

Stańczyk, U. (2015). *Feature Evaluation by Filter, Wrapper, and Embedded Approaches*, pp. 29–44. Berlin, Heidelberg: Springer Berlin Heidelberg. (Cited on page 7.)

Sun, Y., A. Wong, and M. Kamel (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence 23*, 687–719. (Cited on page 10.)

Tantithamthavorn, C., A. E. Hassan, and K. Matsumoto (2018). The impact of class rebalancing techniques on the performance and interpretation of defect prediction models. *IEEE Transactions on Software Engineering*, 1–1. (Cited on page 11.)

Tate, R. F. (1954). Correlation between a discrete and a continuous variable. point-biserial correlation. *Ann. Math. Statist. 25*(3), 603–607. (Cited on page 60.)

Tavallaee, M., N. Stakhanova, and A. Ghorbani (2010). Toward credible evaluation of anomaly-based intrusion-detection methods. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 40*, 516 – 524. (Cited on page 10.)

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B, Methodological 58*(1), 267–288. (Cited on page 8.)

Tsuruoka, Y., J. Tsujii, and S. Ananiadou (2009). Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of*

the AFNLP: Volume 1 - Volume 1, ACL '09, USA, pp. 477–485. Association for Computational Linguistics. (Cited on pages 22 and 60.)

Wang, H., T. M. Khoshgoftaar, and J. Van Hulse (2010). A comparative study of threshold-based feature selection techniques. In *2010 IEEE International Conference on Granular Computing*, pp. 499–504. (Cited on page 31.)

Weisberg, S. (2005). *Applied Linear Regression*. Hoboken: John Wiley Sons, Incorporated. (Cited on pages 8 and 14.)

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin 1*(6), 80–83. (Cited on pages 46 and 81.)

Wu, G. and E. Chang (2003). Class-boundary alignment for imbalanced dataset learning. (Cited on page 10.)

Xia, G. and W. Jin (2008). Model of customer churn prediction on support vector machine. *Systems Engineering - Theory Practice 28*(1), 71–77. (Cited on pages 14, 16, 29 and 57.)

Xue, Y., L. Zhang, B. Wang, Z. Zhang, and F. Li (2018). Nonlinear feature selection using gaussian kernel svm-rfe for fault diagnosis. *Applied intelligence (Dordrecht, Netherlands) 48*(10), 3306–3331. (Cited on page 105.)

Yang, Q. and X. Wu (2006). 10 challenging problems in data mining research. *International Journal of Information Technology Decision Making (IJITDM) 05*(04), 597–604. (Cited on page 10.)

Yang, Z., W. Tang, A. Shintemirov, and Q. Wu (2009). Association rule mining-based dissolved gas analysis for fault diagnosis of power transformers. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 39*, 597 – 610. (Cited on page 10.)

Zhu, B., B. Baesens, A. Backiel, and S. K. L. M. vanden Broucke (2018). Benchmarking sampling techniques for imbalance learning in churn prediction. *Journal of the Operational Research Society 69*(1), 49–65. (Cited on page 10.)

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. journal of the royal statistical society. series b (statistical methodology). *67*(2), 301–320. (Cited on page 8.)

Appendices

Appendix A

A.1 Analysing the behavior of performance measures

To evaluate the best performing machine learning method with RFE, we needed to decide the most suitable feature selection accuracy measure that is obtained using the newly introduced confusion matrix 4.2. Therefore, we derived new confusion matrices with the resulting selected features and calculated their accuracy measures below.

$$Precision_{fs} = \frac{TP_{fs}}{TP_{fs} + FP_{fs}}$$

$$Recall_{fs} = \frac{TP_{fs}}{TP_{fs} + FN_{fs}}$$

$$F1\_score_{fs} = 2\left(\frac{Precision_{fs} * Recall_{fs}}{Precision_{fs} + Recall_{fs}}\right)$$

$$Recall_{fs} = TPR_{fs} = \frac{TP_{fs}}{Total\_number\_of\_informative}$$

$$FPR_{fs} = \frac{FP_{fs}}{Total\_number\_of\_informative}$$

$$TNR_{fs} = \frac{TN_{fs}}{Total\_number\_of\_non\_informative}$$

$$FNR_{fs} = \frac{FN_{fs}}{Total\_number\_of\_non\_informative}$$

$$Correct\%(Balance\_Accuracy(BA)_{fs}) = \frac{TPR_{fs} + TNR_{fs}}{2}$$

$$Total_{fs} = TP_{fs} + TN_{fs} + FP_{fs} + TN_{fs}$$

$$Rand\_Index(Total\_Accuracy)_{fs} = \frac{TP_{fs} + TN_{fs}}{Total_{fs}}$$

$$Arithmetic\_Mean_{fs} = \frac{Precision_{fs} + Recall_{fs}}{2}$$

$$Geometric\_Mean_{fs} = \sqrt{Precision_{fs} * Recall_{fs}}$$

$$Cohen\_Kappa\_Statistic_{fs} = \frac{Total\_Accuracy_{fs} - Random\_Accuracy_{fs}}{1 - Random\_Accuracy_{fs}}$$

Using simulated data, we observed the behavior of each feature selection performance matrix with the changes of the number of informative features, imbalance rate, and the sample size of the data set.

We analyze data for two different cases, first changing the sample size and then changing the number of informative features in the sample. A summary of generating data for two scenarios is shown in table A.1.

Case1: Fixed sample size

For case 1, when the sample size is fixed, we repeated the same procedure for different such performance matrices. Figures A.1, A.2, A.3, A.4 and A.5 show how each performance measure behave differently with different RFE classification methods with and without SMOTE.

Table A.1: Two data generation steps

	Case1: Fix sample size	Case2: Fix number of informative features
sample size	Fixed 1000	Varying 100 to 1000, increasing by 50
number of informative features	Varying 1 to 30 increasing by 1	Fixed 20
class imbalanced levels	[0.5:0.5] , [0.7:0.3], [0.9:0.1]	[0.5:0.5] , [0.7:0.3], [0.9:0.1]
re-balancing technique (smote)	with/ without	with/ without
feature selection method	RFE	RFE
repetitions	100	100

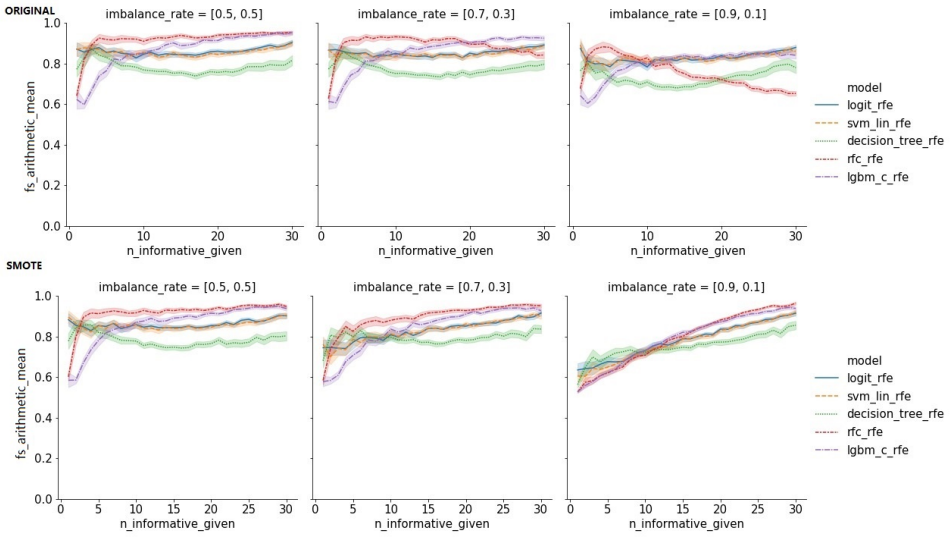

Fig. A.1. Average arithmetic mean$_{fs}$ vs. the number of informative features given by imbalance rate

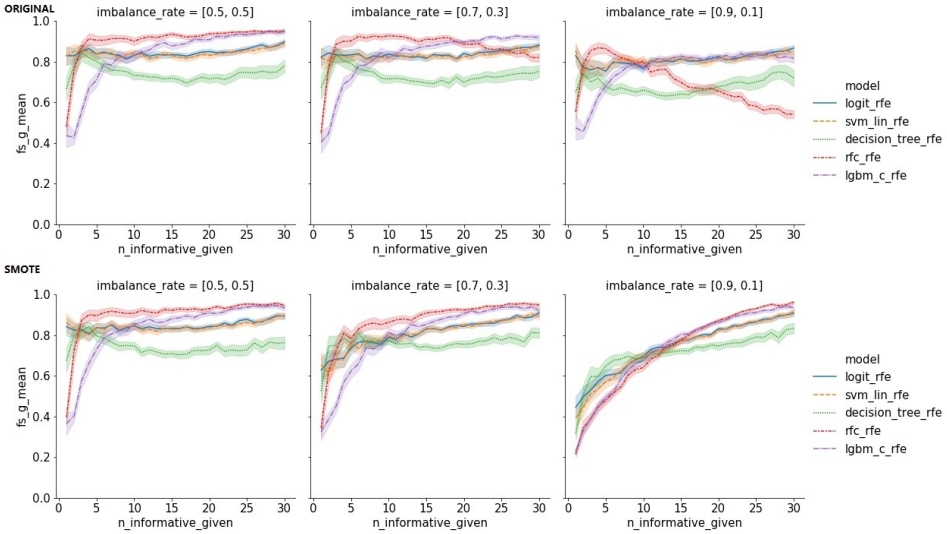

Fig. A.2. Average geometric mean$_{fs}$ vs. the number of informative features given by imbalance rate

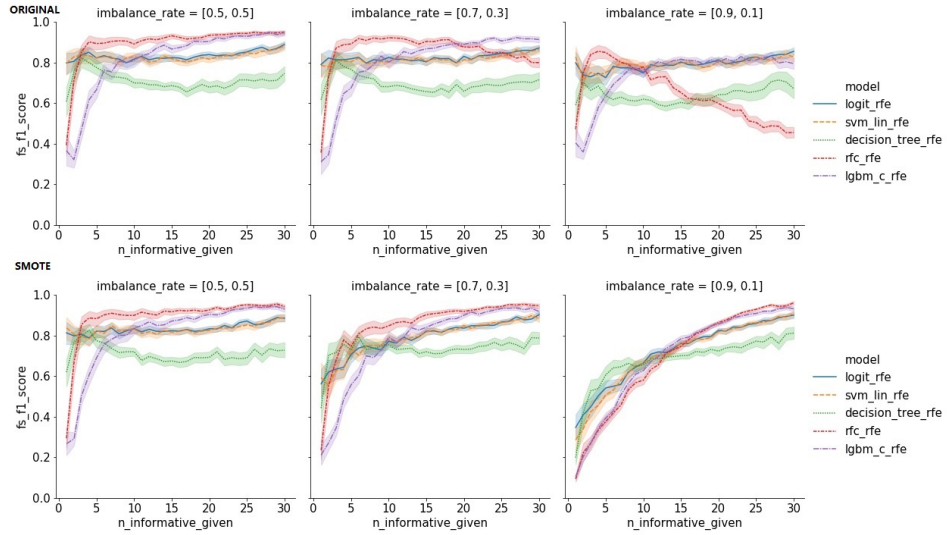

Fig. A.3. Average F1-score$_{fs}$ vs. the number of informative features given by imbalance rate

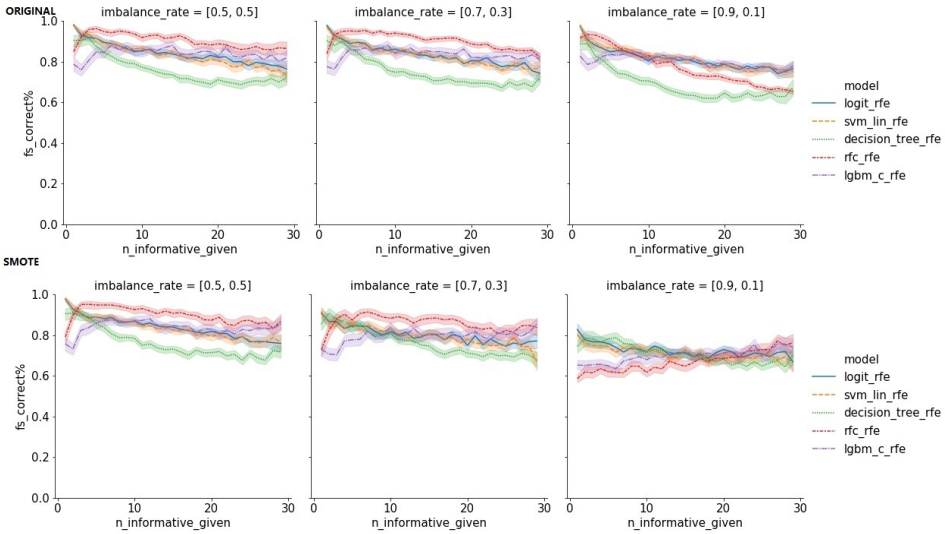

Fig. A.4. Average correct percentage$_{fs}$ vs. the number of informative features given by imbalance rate

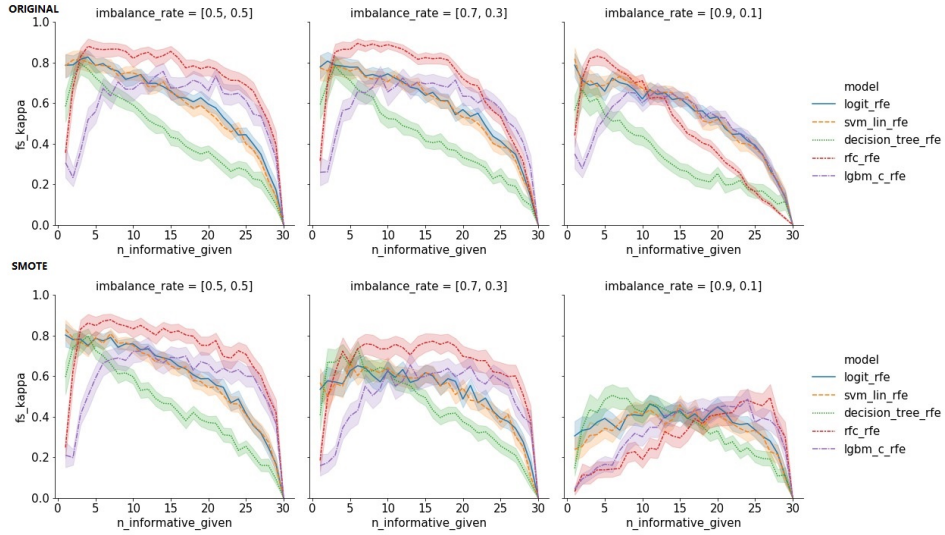

Fig. A.5. Average kappa statistic$_{fs}$ vs. the number of informative features given by imbalance rate

Case2: Fixed number of informative features

For case 2, when the number of informative features is fixed to 30, we repeated the same procedure for different performance matrices. Figure A.6 depicts that there is a noticeable change of the F1-score$_{fs}$ for different sample sizes with different RFE classification methods with and without SMOTE.

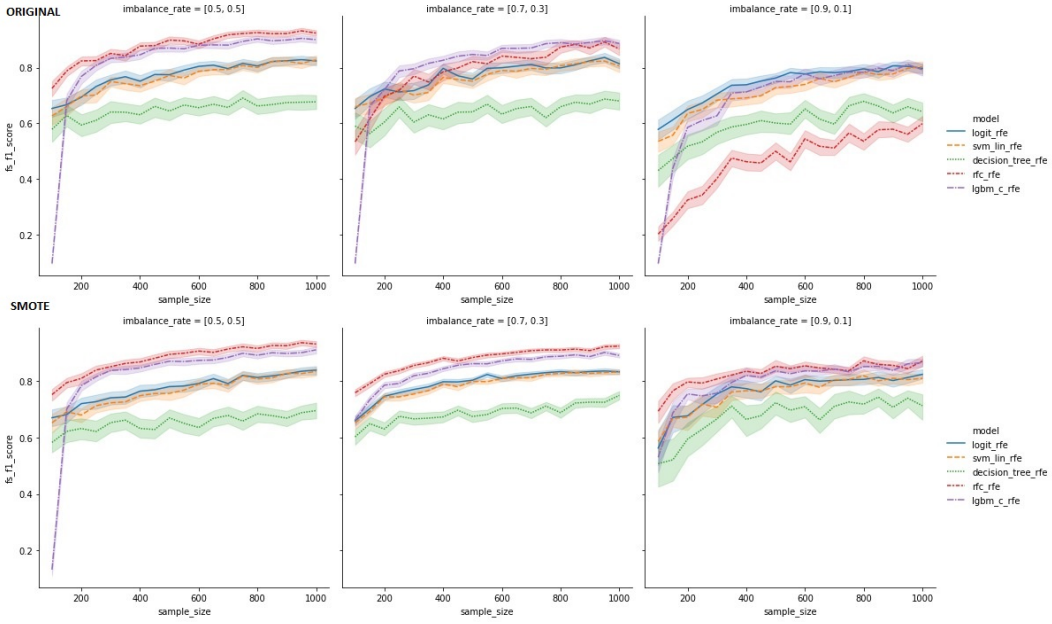

Fig. A.6. Average fs F1-score vs. the sample size by imbalance rate

Appendix B

B.1 Simulation Results for a unified approach for feature selection

To observe the variability of the results in Section 4.4.2 for different classification models, we repeatedly generated 100 data sets for each scenario to meet different practical situations by changing the sample size, the number of informative features, and the class imbalance rate. We applied all four methods explained in above section for each data set and finally calculated the percentage of selecting informative features using the equation 4.1. Fig. B.1 and B.4 show results for the original data whereas Fig. B.7, B.10 and B.13 show results with SMOTE data.

Again to capture the variability of the results in Section 4.4.2, a simulation study was done by applying both PCLFS and Logit-RFE methods on training data with different classification methods. Then we evaluated the prediction results on testing data and recorded the model F1-score and the feature selection correct percentage (Correct$\%_{\text{fs}}$) in both cases. The process repeated 100 times for three different imbalance rates and three different sample sizes.

Fig. B.2 and B.5 show results for the original data whereas Fig. B.8, B.11 and B.14 show results with SMOTE data. According to the figures it can be seen that the PCLFS with different classifiers gives a higher final F1-score in each situation and it works extremely well for small sample size and highly imbalanced data. The figures Fig. B.3, B.6, B.9, B.12 and B.14 imply that the PCLFS method with different classifiers obtains a higher feature selection correct percentage in each situation when the number of informative features in the sample is greater than four.

B.1.1 Without SMOTE

Decision Trees

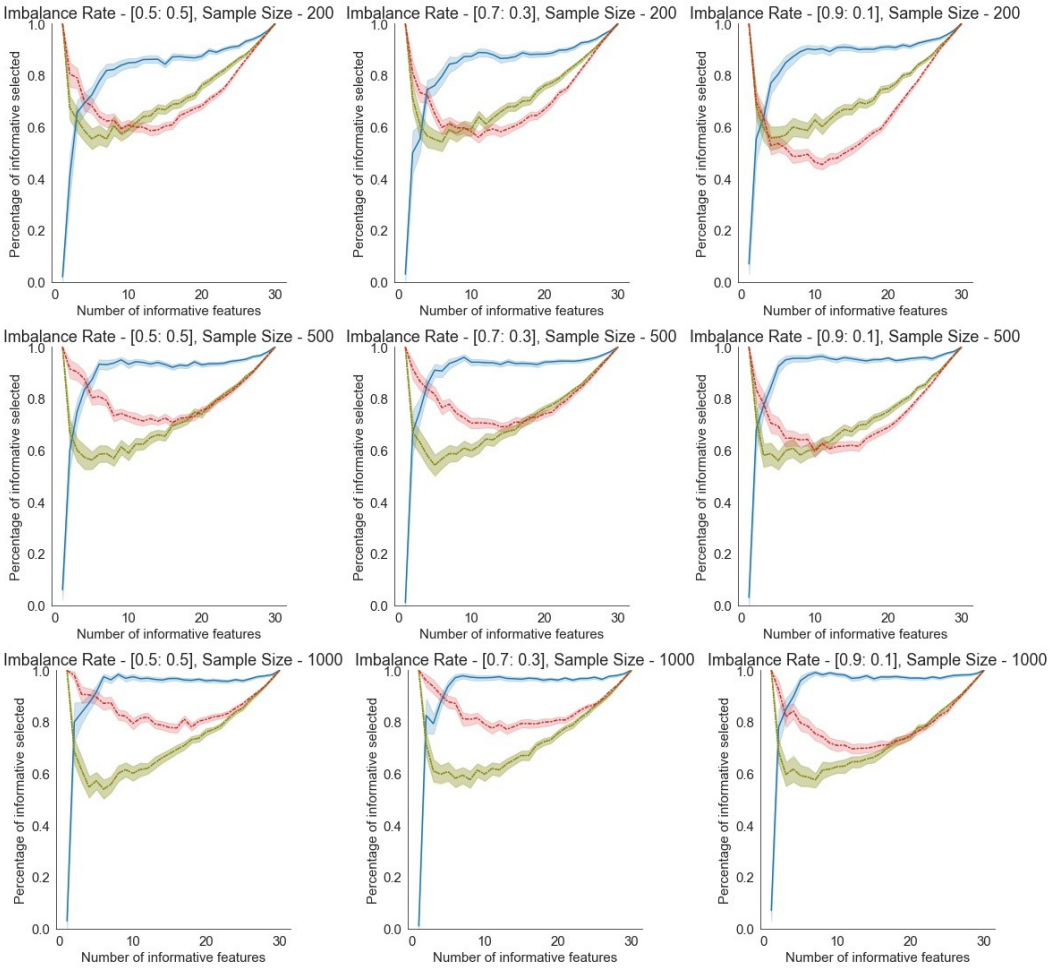

Fig. B.1. Mean percentages of informative features selected by each ordering technique in different class imbalanced levels and sample sizes for the Decision Tree model without SMOTE. The blue line represents the sum of the absolute values of principal component loadings; the red dashed line indicates Decision Tree model-based feature importance results. The overlapped green and orange dashes lines show the absolute correlation and the ANOVA F value classification results.

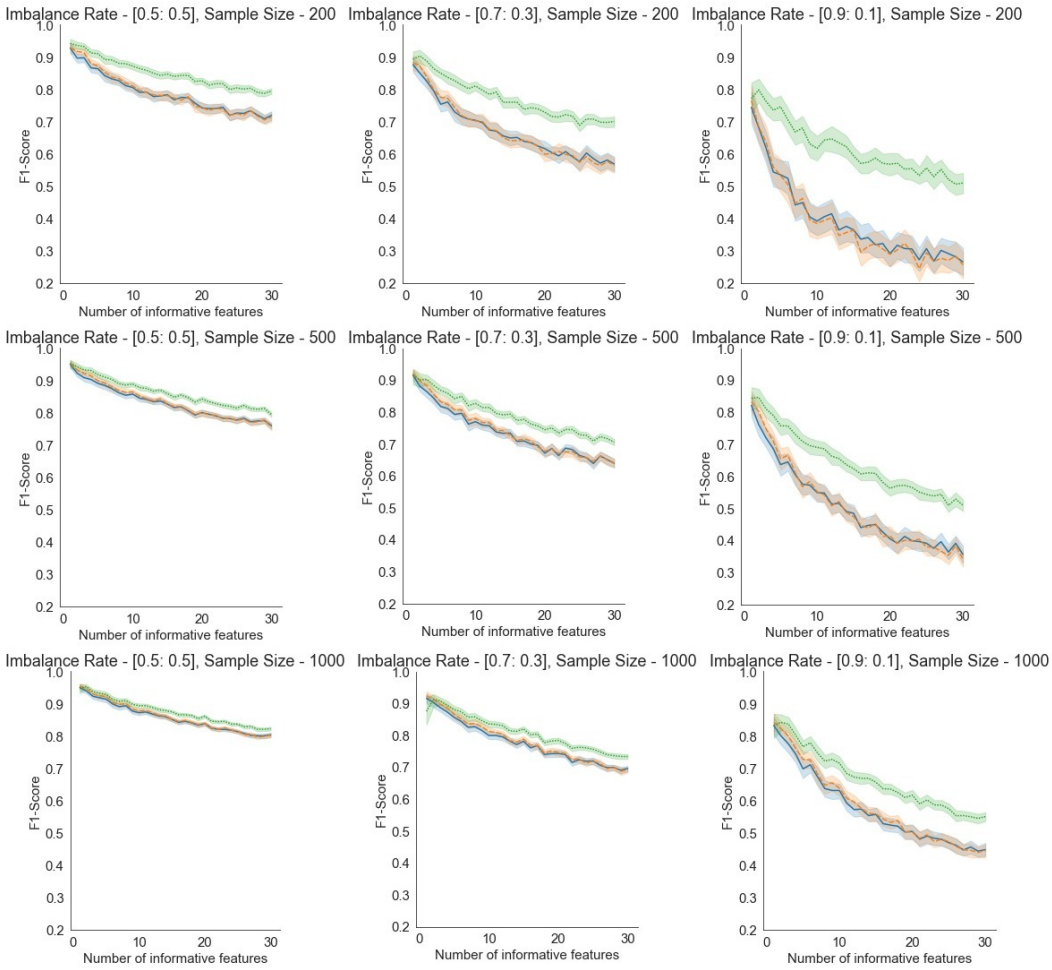

Fig. B.2. Final model F1-scores for the Decision Tree model without SMOTE. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

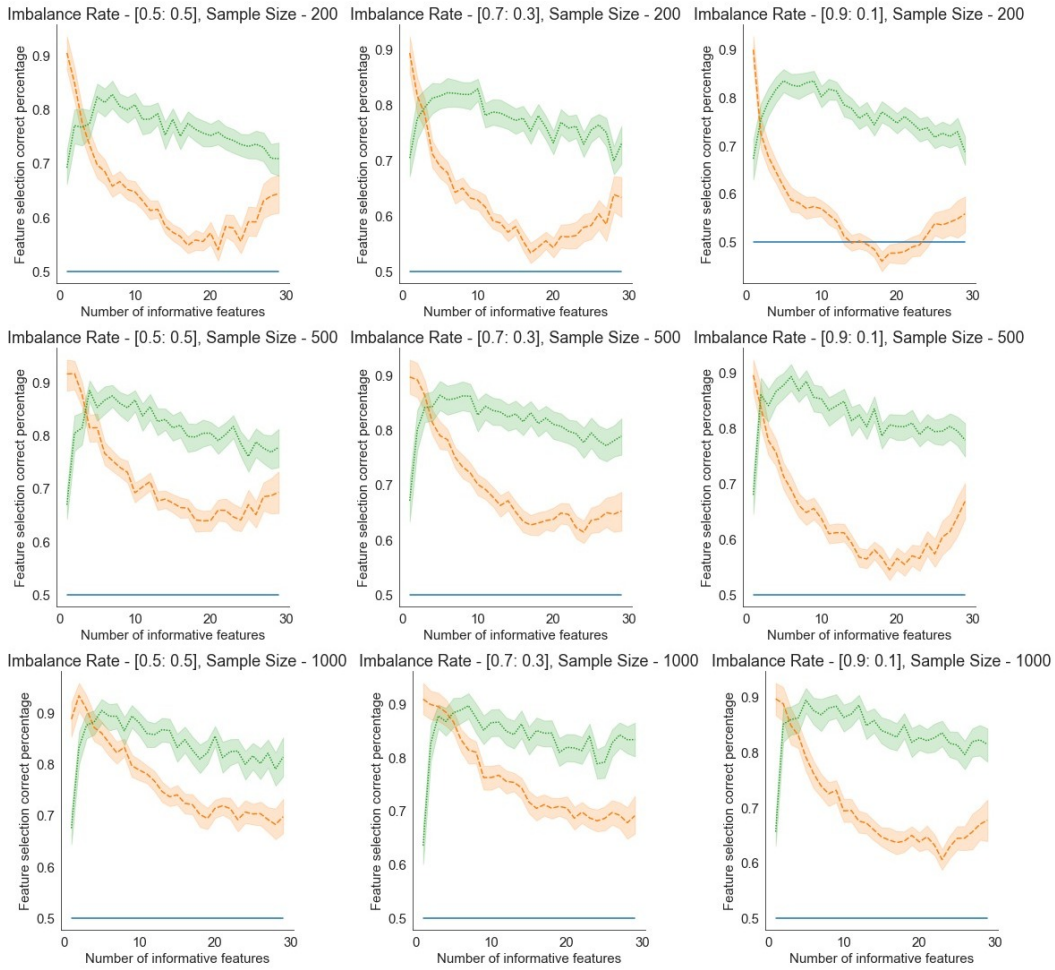

Fig. B.3. Feature selection correct percentages for the Decision Tree model without SMOTE. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

Light Gradient Boosting (LGBM)

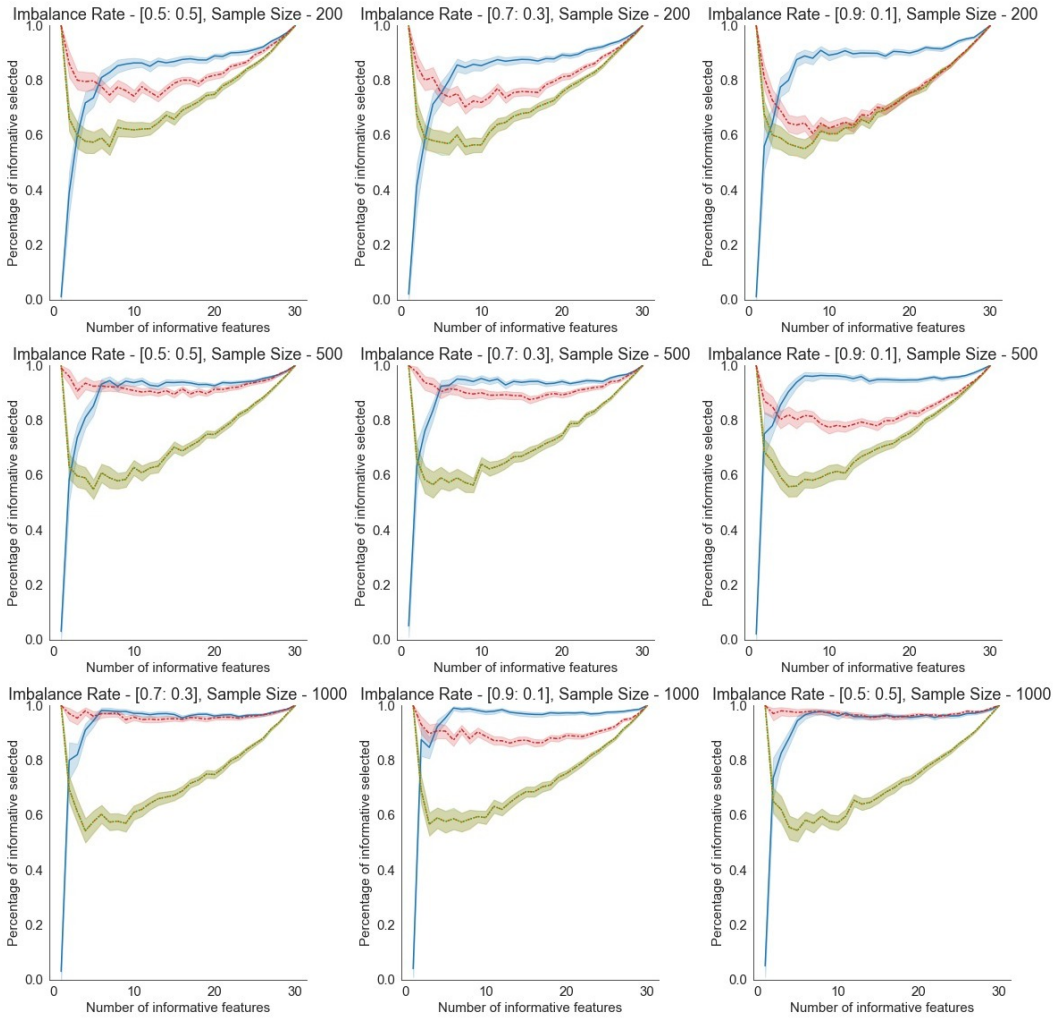

Fig. B.4. Mean percentages of informative features selected by each ordering technique in different class imbalanced levels and sample sizes for the LGBM model without SMOTE. The blue line represents the sum of the absolute values of principal component loadings; the red dashed line indicates LGBM model-based feature importance results. The overlapped green and orange dashes lines show the absolute correlation and the ANOVA F value classification results.

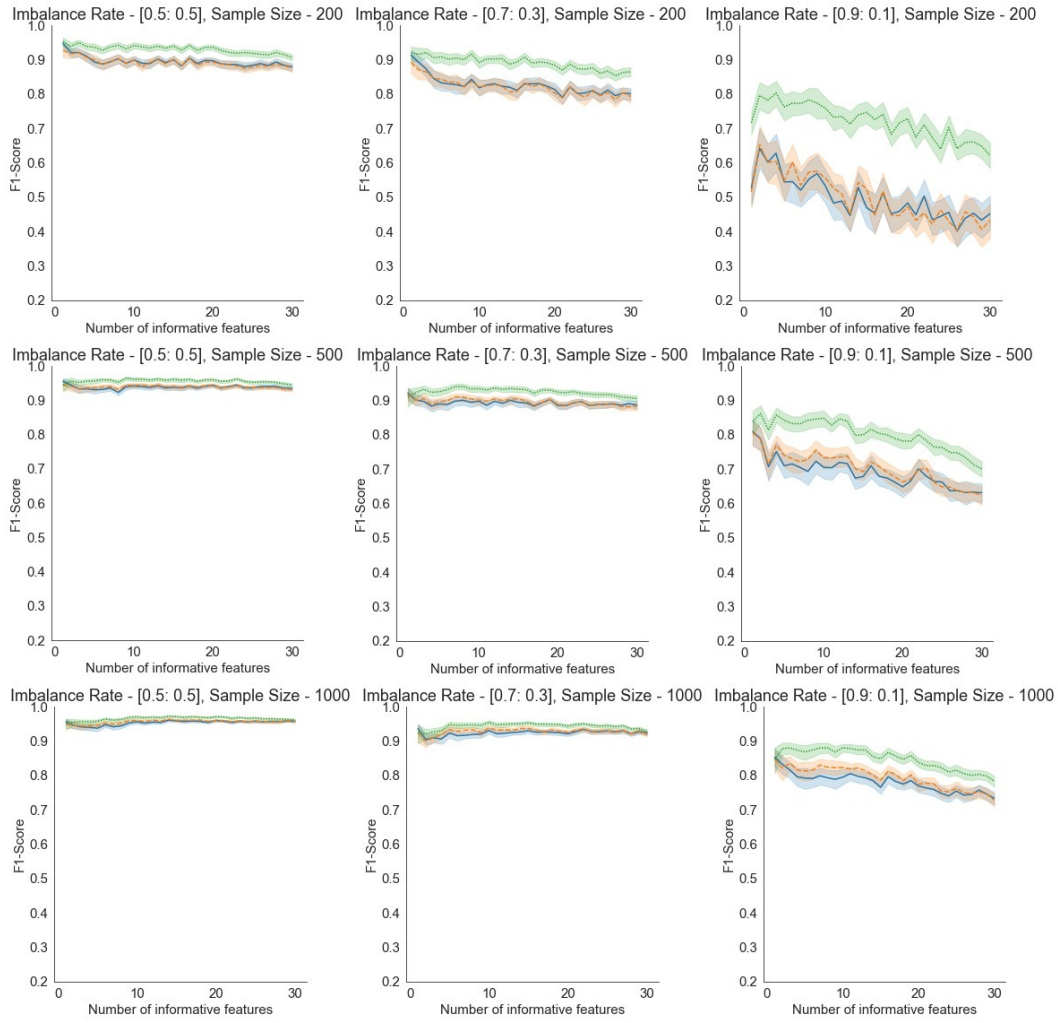

Fig. B.5. Final model F1-scores for the LGBM model without SMOTE. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

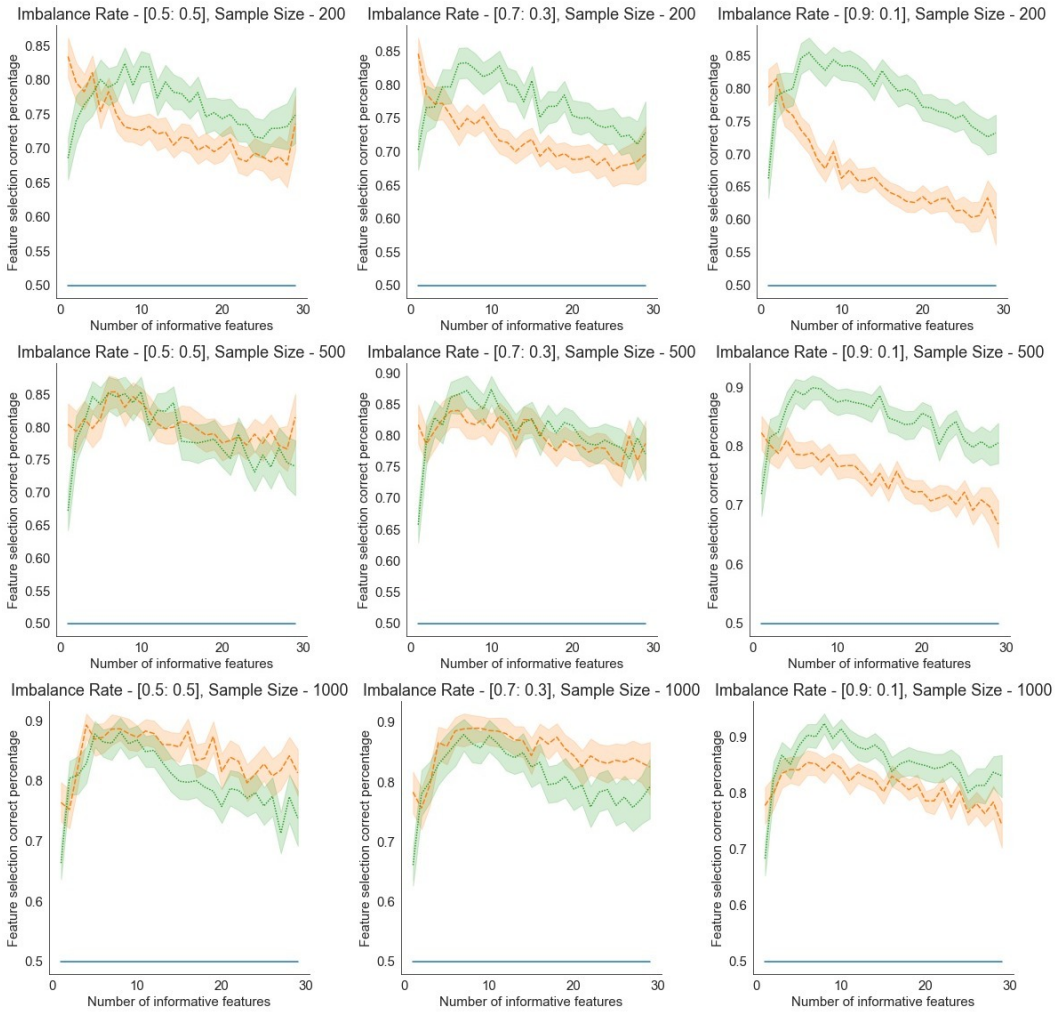

Fig. B.6. Feature selection correct percentages for the LGBM model without SMOTE. The blue line represents the sum of the absolute values of principal component loadings; the red dashed line indicates LGBM model-based feature importance results. The overlapped green and orange dashes lines show the absolute correlation and the ANOVA F value classification results.

B.1.2 With SMOTE

Logistic Regression

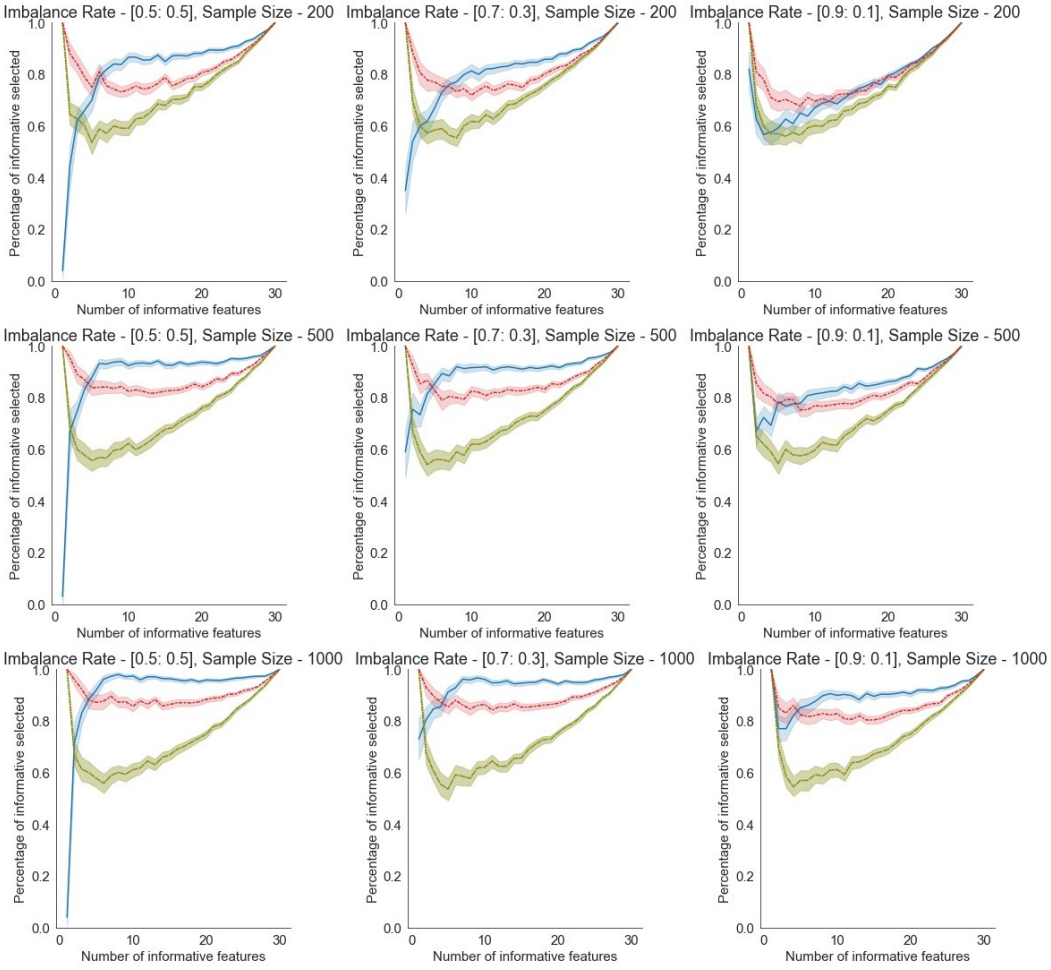

Fig. B.7. Mean percentages of informative features selected by each ordering technique in different class imbalanced levels and sample sizes for the Logit model with SMOTE. The blue line represents the sum of the absolute values of principal component loadings; the red dashed line indicates Logit model-based feature importance results. The overlapped green and orange dashes lines show the absolute correlation and the ANOVA F value classification results.

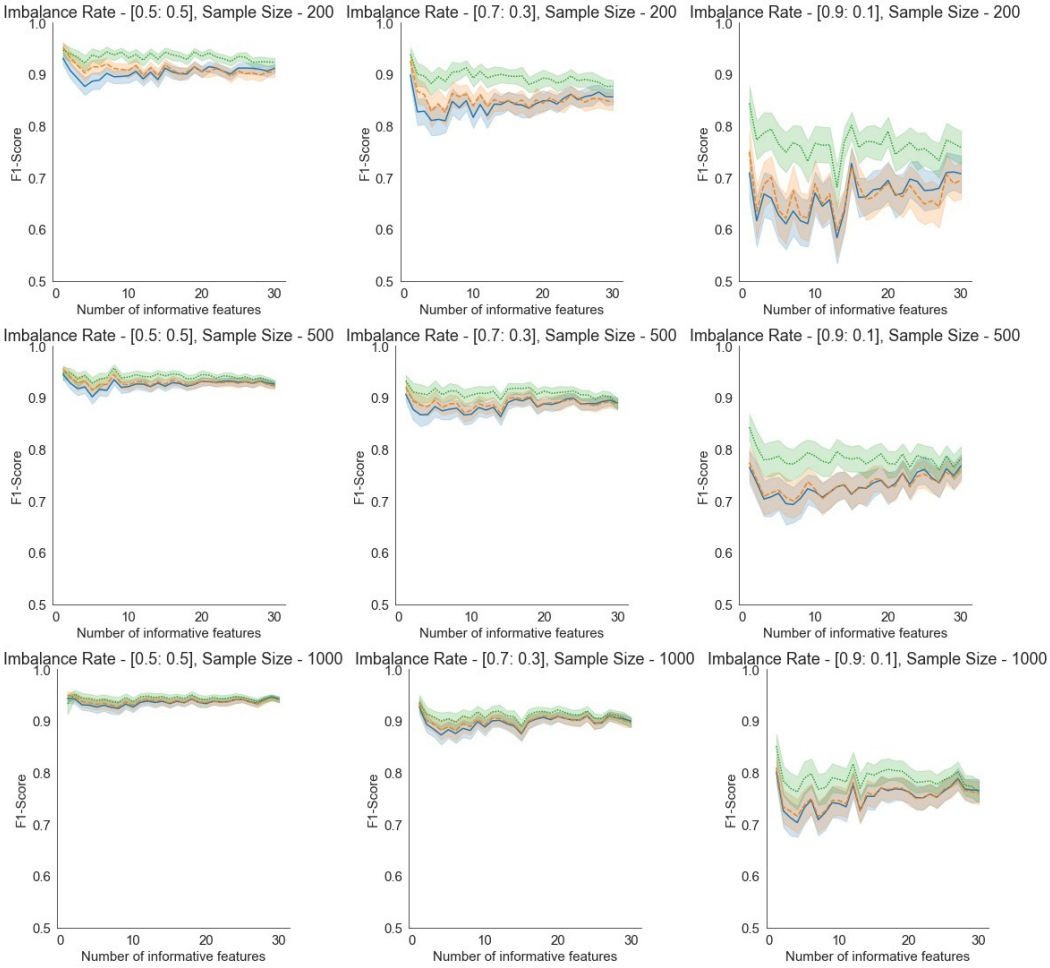

Fig. B.8. Final model F1-scores for the Logit model with SMOTE. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

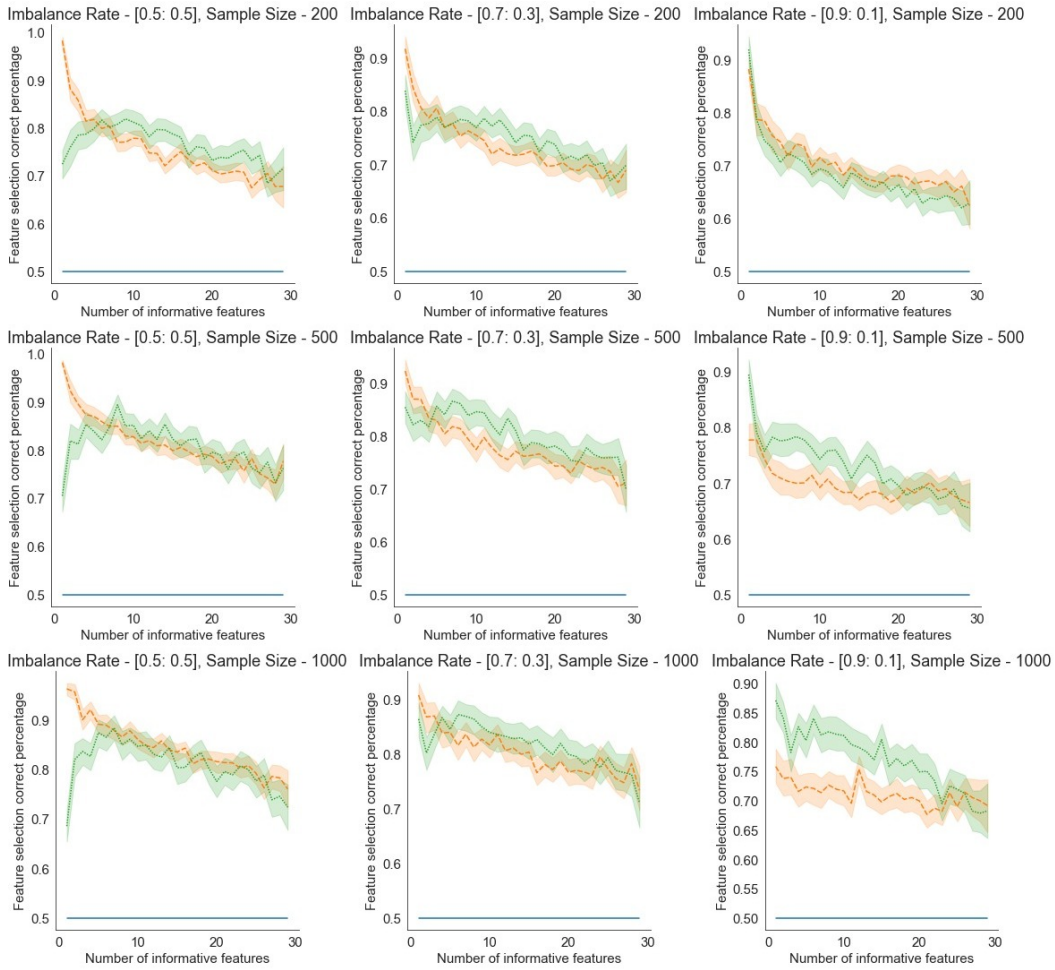

Fig. B.9. Feature selection correct percentages for the Logit model with SMOTE. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

Decision Trees

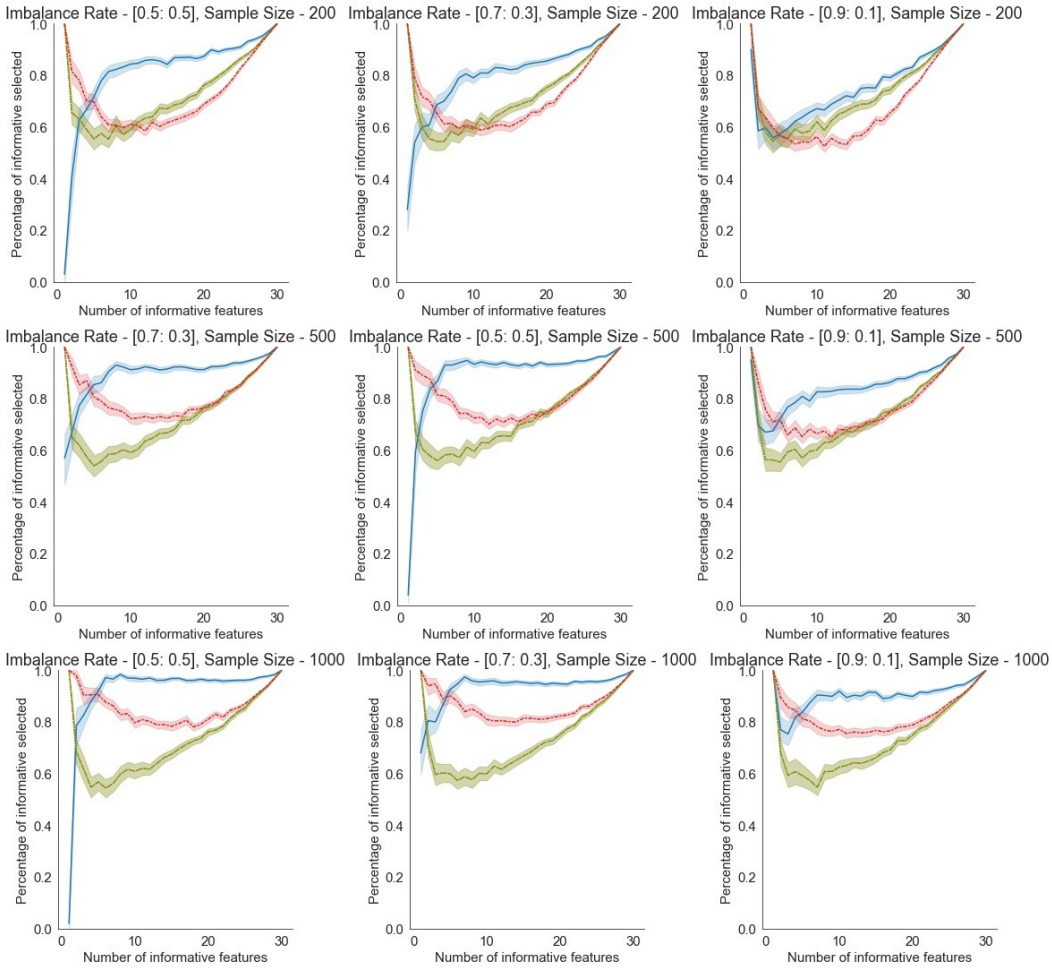

Fig. B.10. Mean percentages of informative features selected by each ordering technique in different class imbalanced levels and sample sizes for the Decision Tree model with SMOTE. The blue line represents the sum of the absolute values of principal component loadings; the red dashed line indicates Decsion Tree model-based feature importance results. The overlapped green and orange dashes lines show the absolute correlation and the ANOVA F value classification results.

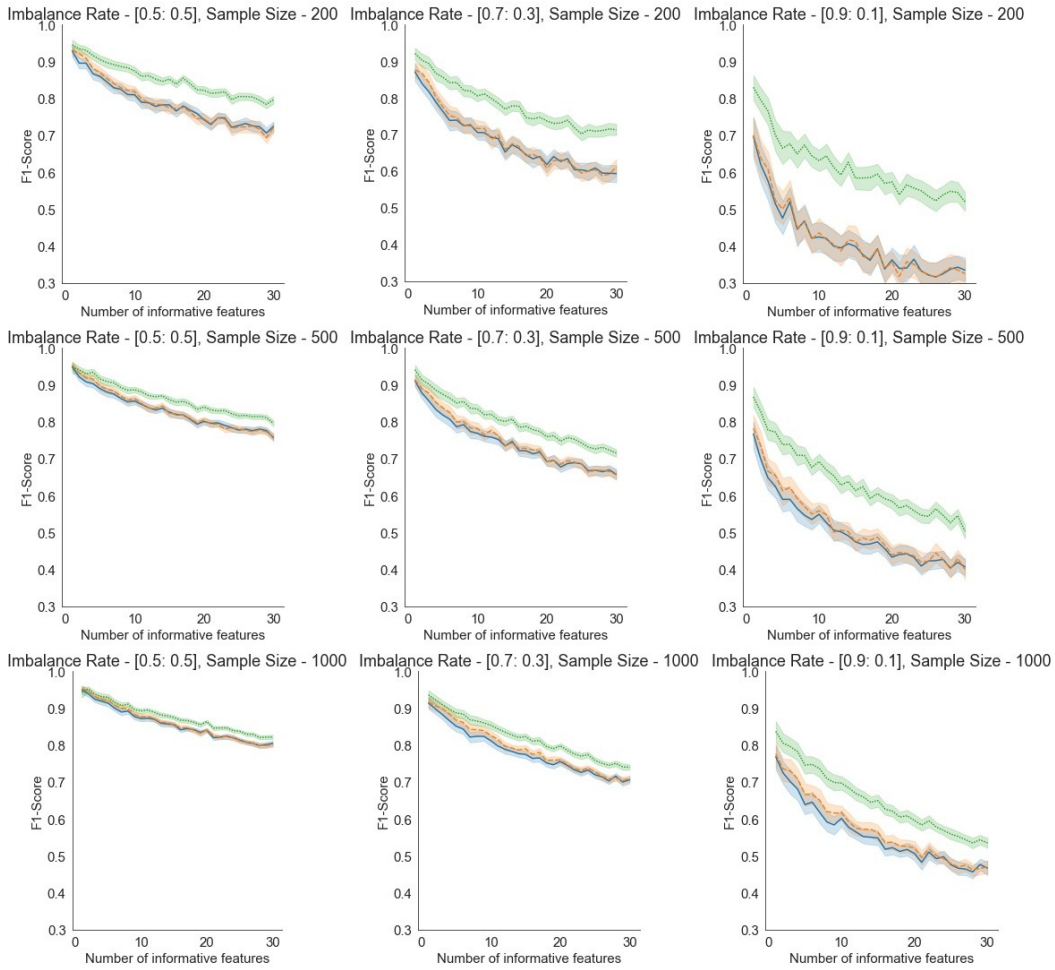

Fig. B.11. Final model F1-scores for the Decision Tree model with SMOTE. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

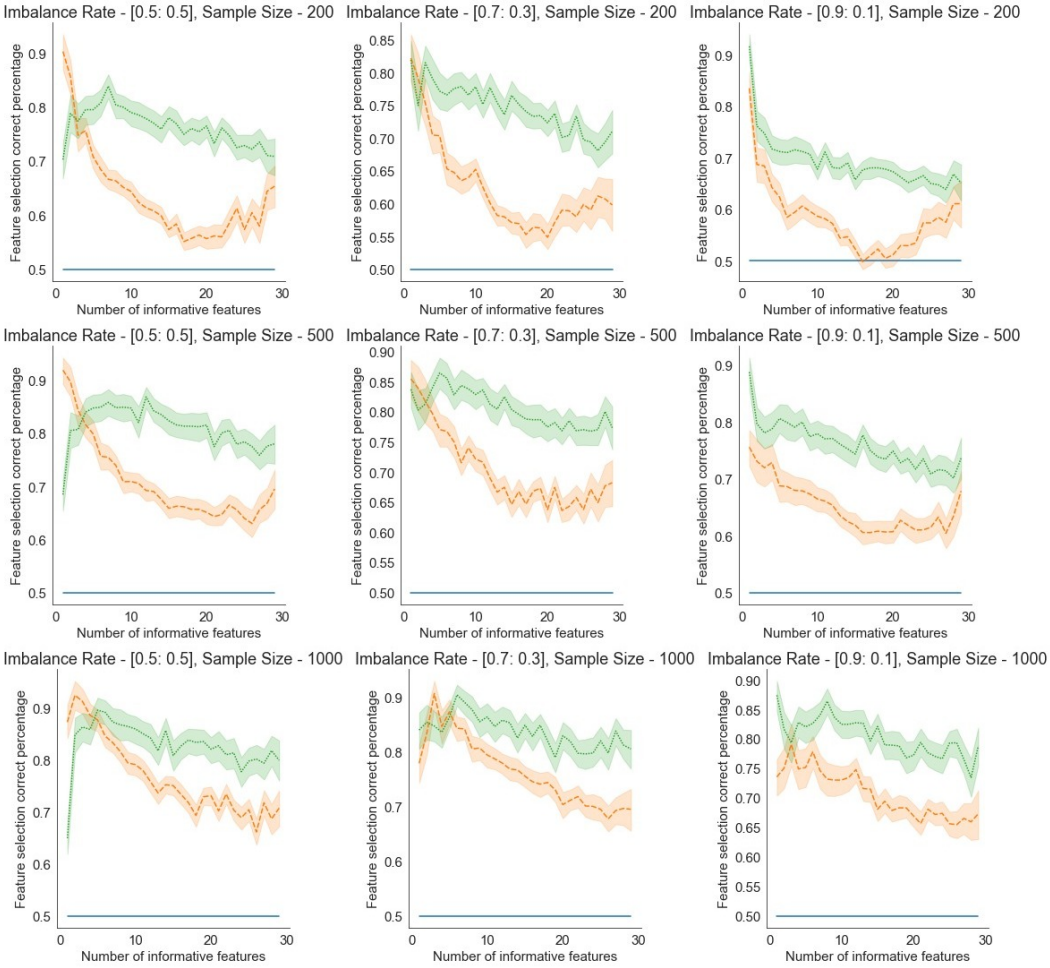

Fig. B.12. Feature selection correct percentages for the Decision Tree model with SMOTE. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

Light Gradient Boosting (LGBM)

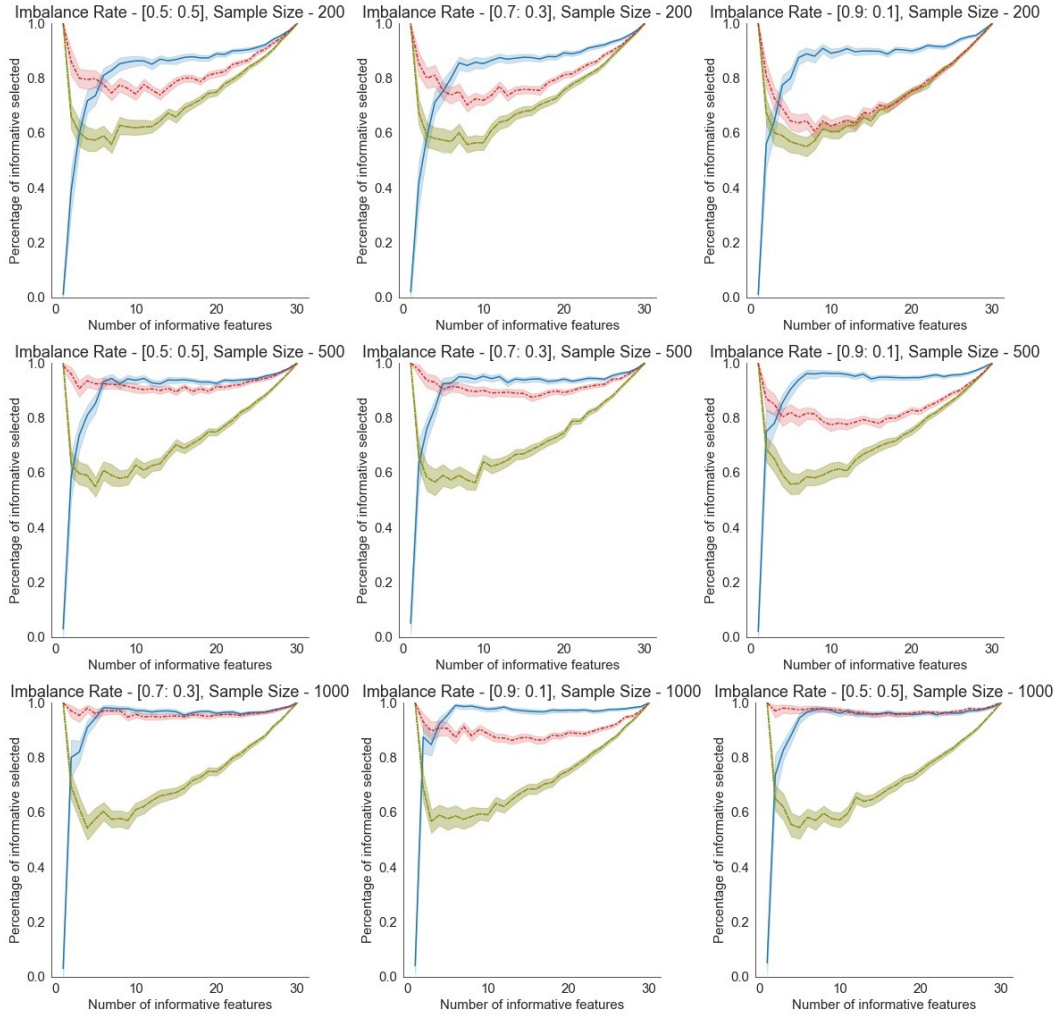

Fig. B.13. Mean percentages of informative features selected by each ordering technique in different class imbalanced levels and sample sizes for the LGBM model with SMOTE. The blue line represents the sum of the absolute values of Principal component loadings; the red dashed line indicates the ANOVA F value classification results. The overlapped green and orange dashes lines show the absolute correlation and logit model-based feature importance results.

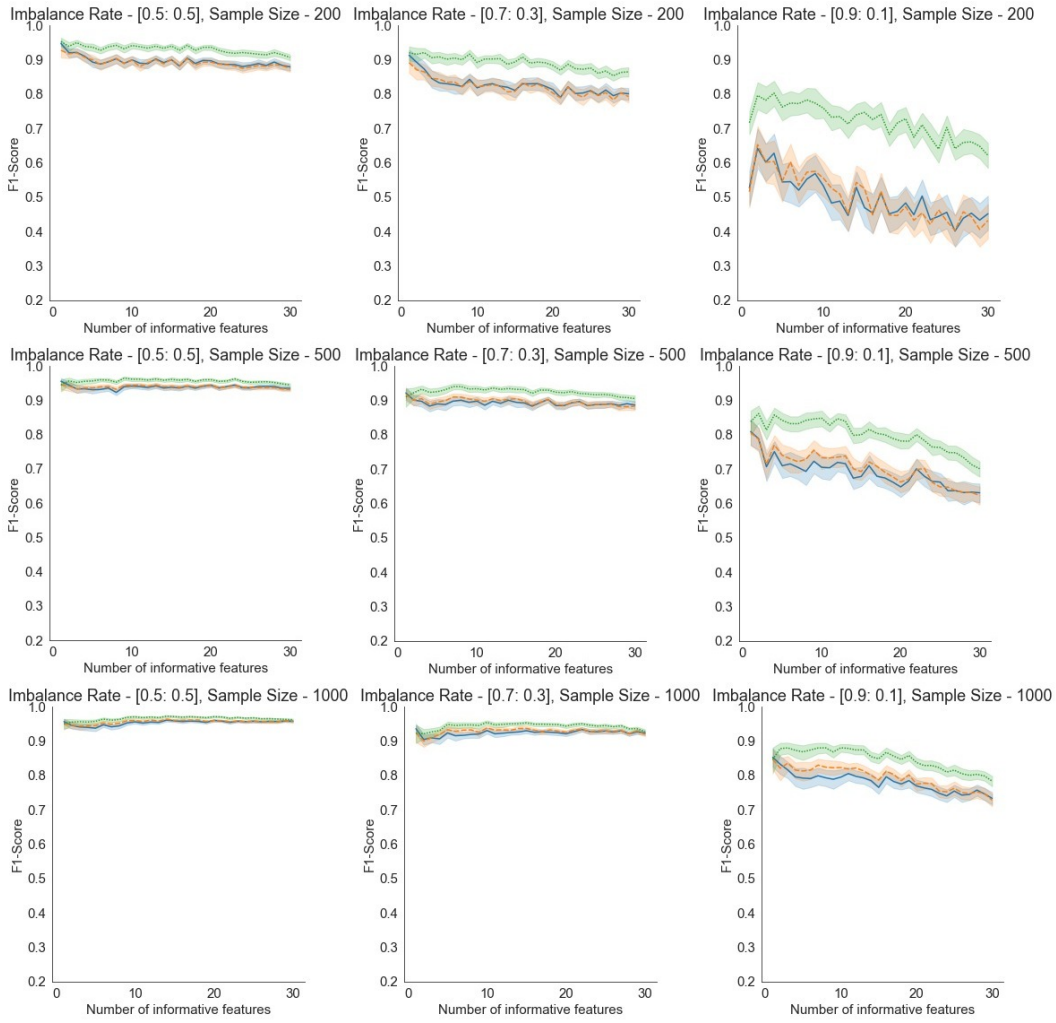

Fig. B.14. Final model F1-scores for the LGBM model with SMOTE. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

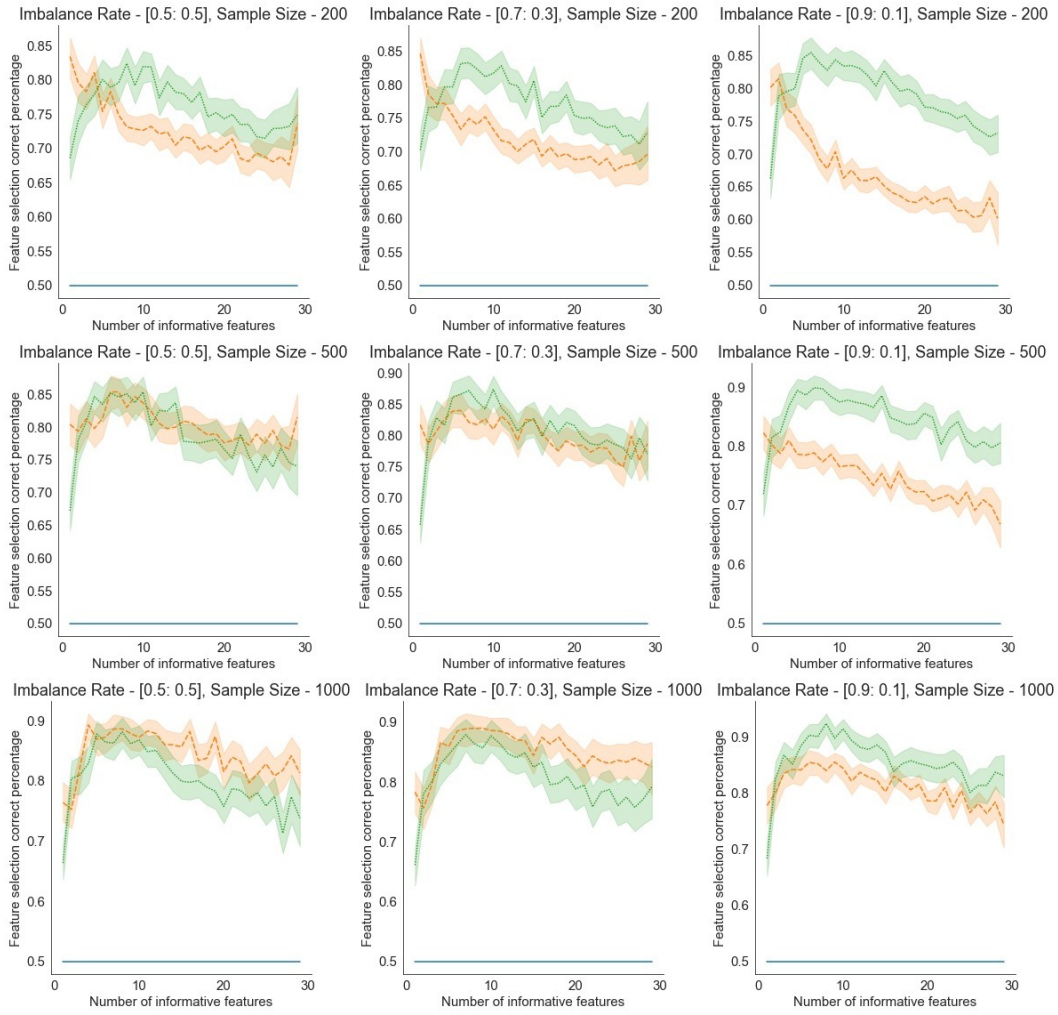

Fig. B.15. Feature selection correct percentages for the LGBM model with SMOTE. The green line represents the PCLFS method; the orange dashed line indicates the RFE results, while the blue line shows the baseline model results without feature selection.

Appendix C

C.1 Results of the Combination

Referring to the 5.2, Fig. C.1, C.2, C.3 and C.4 present the results of the comparison of RFE, PCLFS, and PCLFS-ext methods for other classification models, with highly imbalanced data with 90%:10% rate and a sample size of 1000.

Light Gradient Boosting (LGBM)

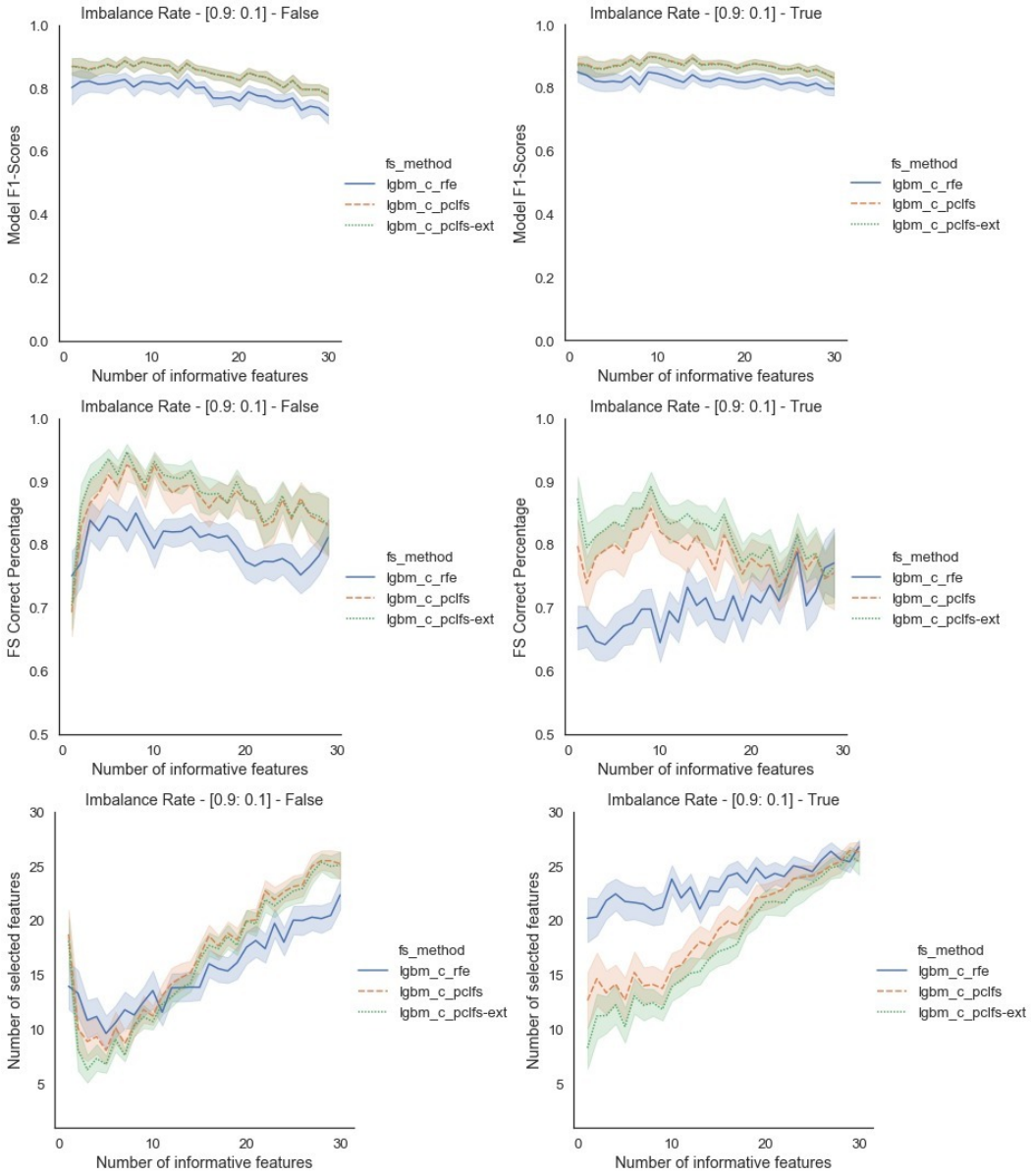

Fig. C.1. Rows represent final F1-scores, Feature selection correct percentages, and the number of informative selected features, whereas the right-hand side column with original data and left is with SMOTE data for the Lgbm_C classifier with a threshold of 0.0017.

Decision Trees

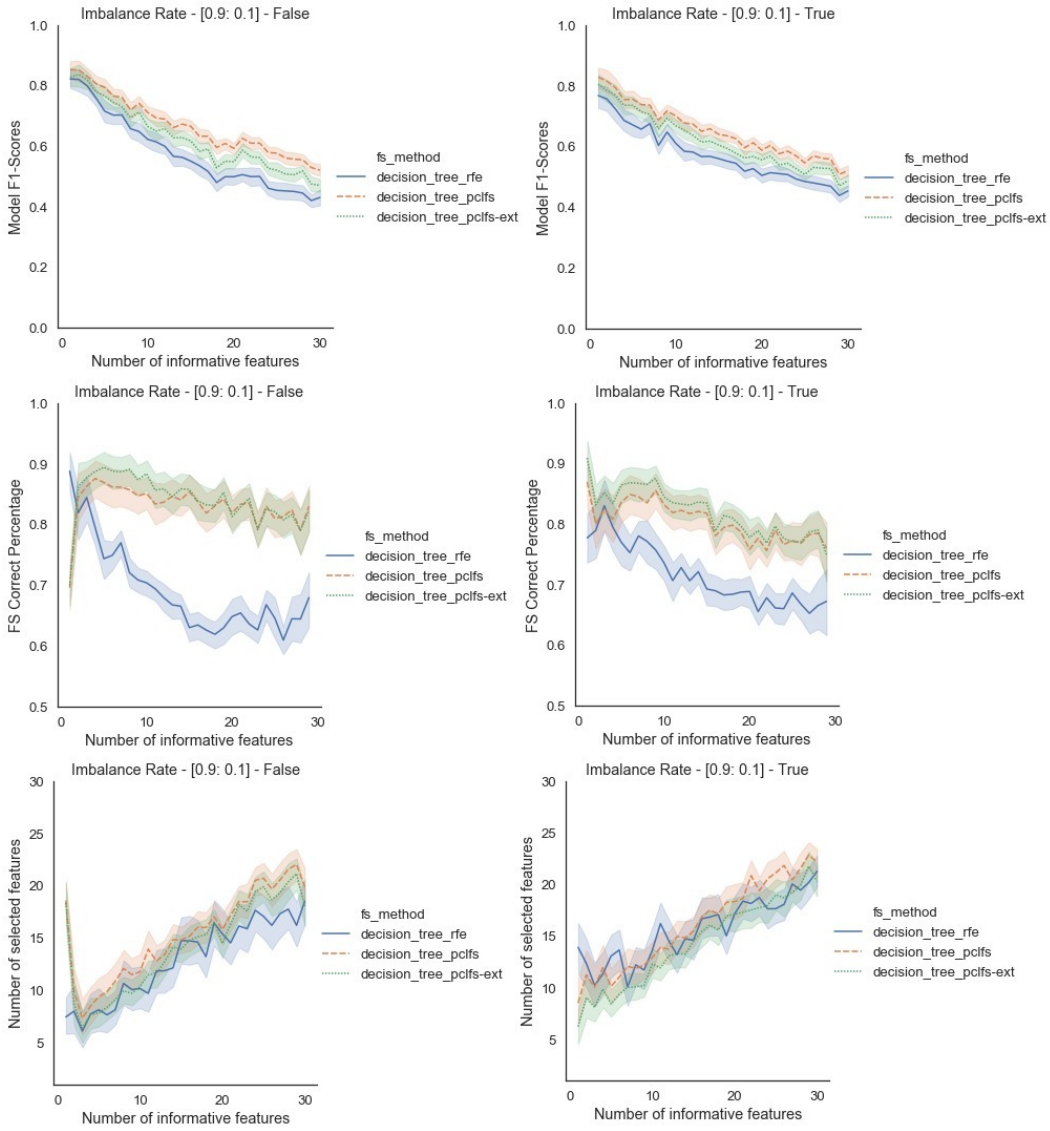

Fig. C.2. Rows represent final F1-scores, Feature selection correct percentages, and the number of informative selected features, whereas the right-hand side column with original data and left is with SMOTE data for the Decision tree classifier with a threshold of 0.0017.

Random Forest Classifier

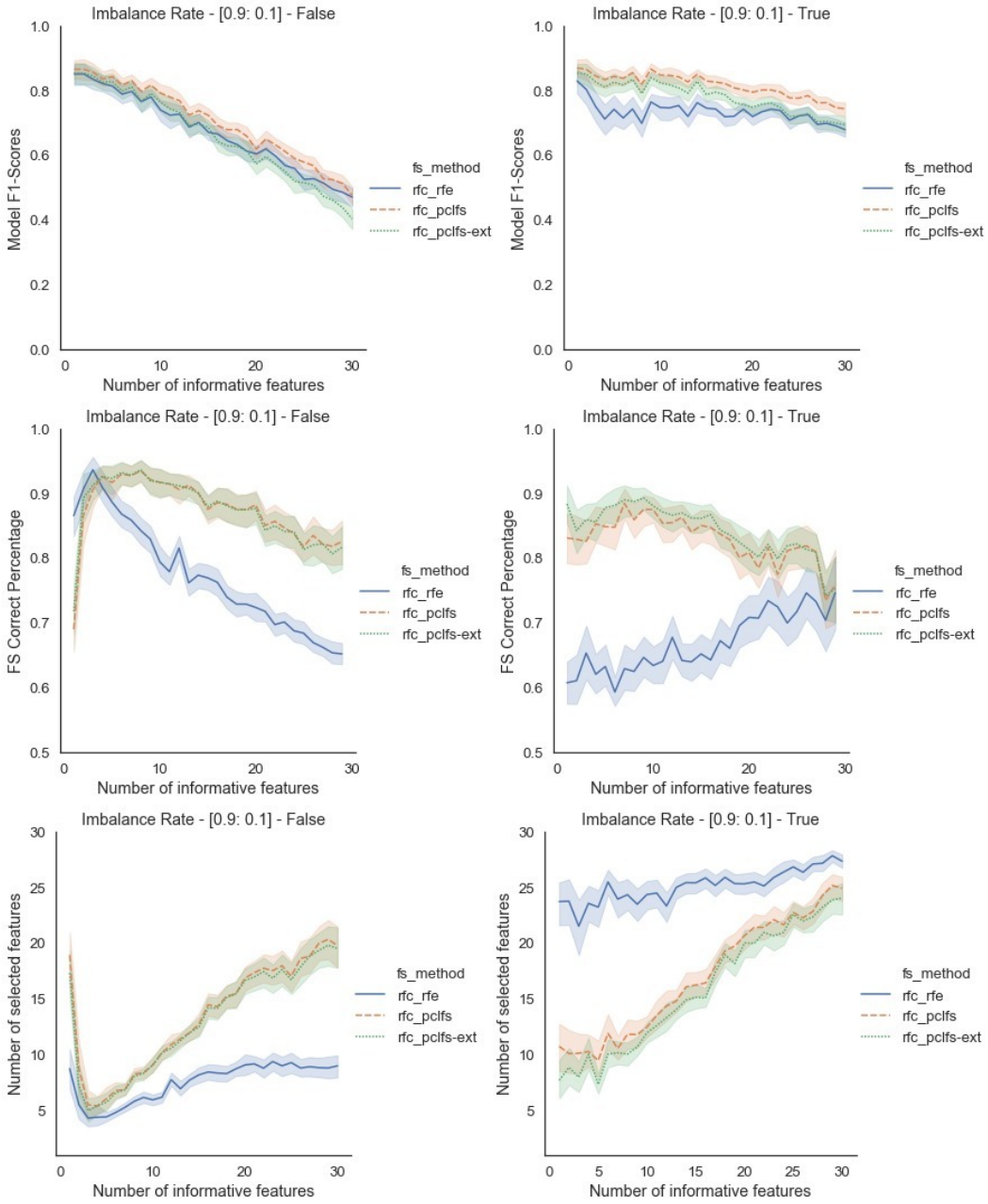

Fig. C.3. Rows represent final F1-scores, Feature selection correct percentages, and the number of informative selected features, whereas the right-hand side column with original data and left is with SMOTE data for the RFC with a threshold of 0.0017.

SVM-Linear

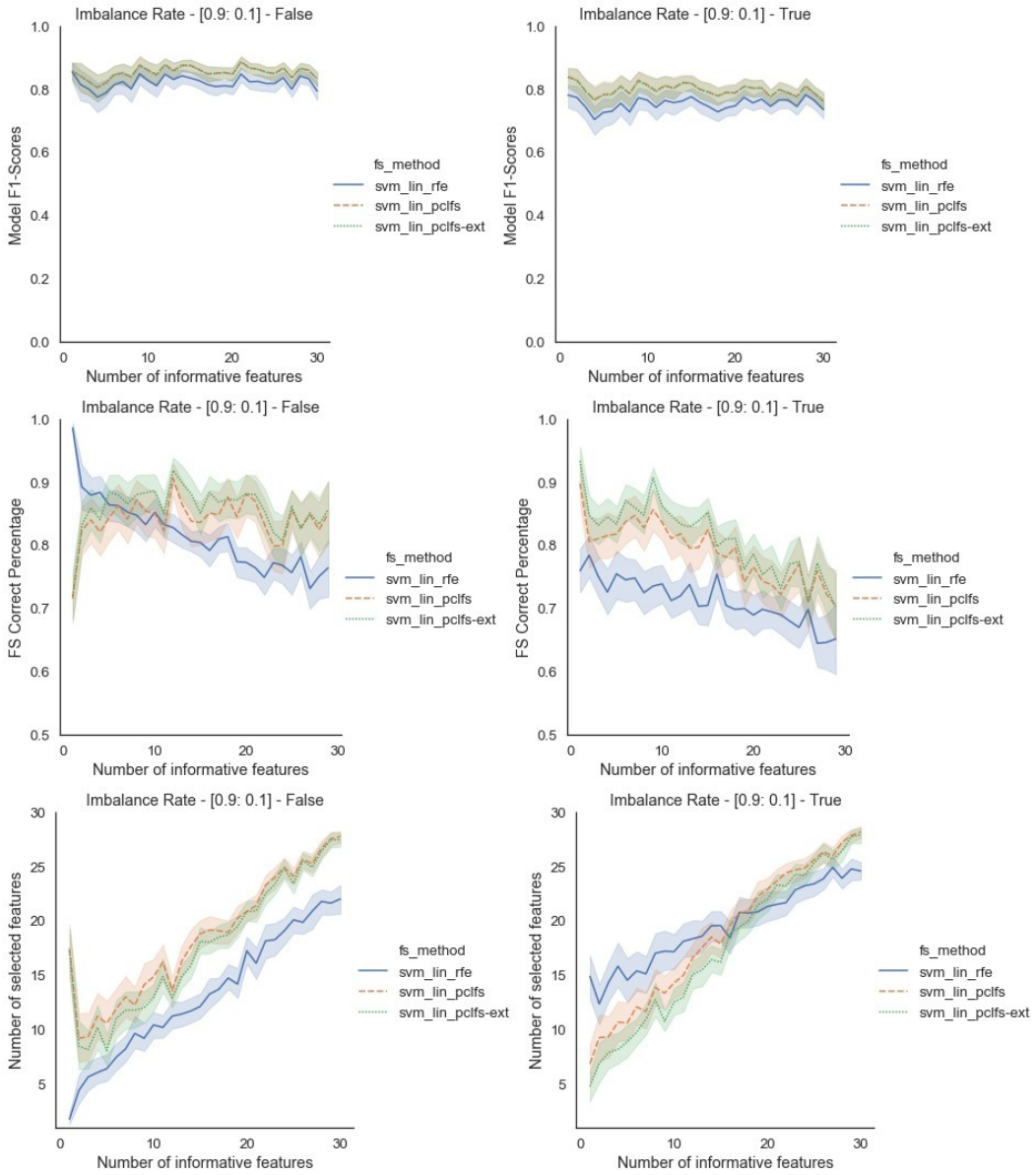

Fig. C.4. Rows represent final F1-scores, Feature selection correct percentages, and the number of informative selected features, whereas the right-hand side column with original data and left is with SMOTE data for the SVM-linear classifier with a threshold of 0.0017.