Machine Learning Approaches for Predicting Bank Customer Subscription: A Comparative Analysis

Zibo Zhao*

Electronic Information School, Wuhan University, 430072, Wuhan, China

Abstract. As people enter the big data era, the traditional banking industry faces huge competitive pressure from new Internet financial products, which requires the traditional banking industry to use data mining and machine learning methods to optimize marketing strategies. Based on the bank marketing data set, this paper explores the effectiveness of several machine learning methods in predicting potential customers, including random forest, k-nearest neighbor (KNN) algorithm, and logistic regression, and conducts comparative analysis before and after data oversampling. Experiments show that the synthetic minority oversampling technique can effectively strengthen the model's capacity to recognize minority samples, but it may cause overfitting. Among them, the random forest has the best overall performance; logistic regression is limited by its linear assumption and performs slightly worse; the KNN algorithm is sensitive to noise and unbalanced data and has poor results. Future research can explore combining different sampling methods or using model integration to improve performance. This study provides important reference significance for the banking industry in accurately positioning potential customers.

1 Introduction

With the widespread application of information technology and big data analysis in the financial field, the banking industry faces fierce competition from new Internet financial products. To meet this challenge, traditional banks need to utilize big data technology to accurately tap potential customer groups, improve marketing efficiency, and reduce costs.

At present, various new financial products based on the Internet platform are constantly emerging. Compared with the traditional bank's fixed deposit business, their convenience and open financial management space have attracted a large number of customers to invest. In response to competition from Internet financial products, the traditional banking industry must leverage data analysis technologies, such as machine learning, to effectively identify potential customers within vast amounts of customer information, enhance marketing outcomes, and cut costs.

Analyzing data to discover potential customers based on massive customer information is a classification task. For this type of decision-making problem, Palaniappan et al. [1] used

* Corresponding author: zibozhao@whu.edu.cn

three methods: naive Bayes, random forest, and decision tree to conduct modeling analysis, and found through experiments that the random forest algorithm has better accuracy in classification prediction. Lawi et al. [2] used the Adaboost integrated Support Vector Machine (SVM) model to predict potential customers. Overall, it is necessary to use machine learning methods to analyze bank marketing data and establish a classification model to accurately find potential users. This can not only reduce marketing costs but also obtain greater benefits from the target customers discovered.

This paper aims to explore the effects and differences of different machine learning algorithms in the task of bank customer classification prediction. In the experiment, three models, logistic regression, k-nearest neighbor (KNN) algorithm and random forest, were used to fit and predict the data sets before and after oversampling, and a comparative analysis was conducted based on various evaluation indicators.

2 Data and methods

2.1 Dataset Introduction

The data used in this paper comes from the UCI website (Bank Marketing - UCI Machine Learning Repository) [3]. The data is related to the direct marketing campaigns of a Portuguese banking institution. The purpose of the study is to make a prediction on whether the customer will subscribe to the bank term deposit. The data set has a total of 45,212 examples, including 17 variables, with 16 features and 1 label. The specific content is shown in Table 1.

Table 1. Variables and descriptions.

Variable Name	Type	Description
age	int64	age
job	object	type of job
marital	object	marital status
education	object	education Level
default	object	Is credit in default?
balance	int64	average yearly balance
housing	object	has a housing loan?
loan	object	has a personal loan?
contact	object	contact communication type
day	int64	number of days of continuous contact in a month
month	object	last contact month of the year
duration	int64	last contact duration, in seconds
campaign	int64	number of contacts performed during this campaign and for this client
pdays	int64	number of days that passed by after the client was last contacted from a previous campaign
previous	int64	number of contacts performed before this campaign and for this client
poutcome	object	outcome of the previous marketing campaign
y	object	has the client subscribed to a term deposit?

Among the 16 features shown in Table 1, ‘age’, ‘job’, ‘marital’, and ‘education’ are the basic information of customers; ‘loan’, ‘balance’, ‘housing’, and ‘default’ are the financial information of customers; and the remaining features are marketing-related information.

The label of the data set is ‘y’, which means whether the customer subscribes to a time deposit. It is essentially a binary classification problem.

2.2 Data preprocessing

There may be missing values, outliers, etc. in the original data, which cannot be directly analyzed. In addition, there are many variables in the data set, and the values of each variable have not been formatted or standardized, which makes it inconvenient to directly build a model and train it. Therefore, it is necessary to preprocess the original data set, clean, transform, and format the original data to make it more suitable for model training and analysis, thereby enhancing the performance of the model.

First, this paper conducts data cleaning. By counting the missing values in the original data set, it is found that there are no missing values in the numerical features, and most of the categorical features have no missing values. Only four variables, ‘poutcome’, ‘contact’, ‘job’, and ‘education’, have "unknown" values. Among the categorical variables, the "unknown" values for ‘poutcome’ represent more than 80% of the total, making it impractical to fill or delete these entries. Thus, the "unknown" values for ‘poutcome’ are treated as valid. In contrast, the "unknown" values for ‘job’ and ‘education’ account for only 0.64% and 4.1%, respectively, allowing for the direct deletion of those samples. Additionally, the "unknown" value for ‘contact’ comprises 28.8% of the dataset; therefore, this paper employs the random forest method to predict and fill in these samples [4].

Furthermore, the data for this study is divided by randomly extracting 20% for the test set, leaving 80% for the training set, which includes 34,553 samples.

Next, this study conducts a correlation analysis of the seven numerical features in the training data set. This paper uses the Pearson correlation coefficient analysis method to measure the correlation between variables [5]. Assume that there are two variables X and Y, and the calculation of their Pearson correlation coefficient r is shown in formula (1):

$$r = \frac{\sum_{i=1}^{n}(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\overline{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\overline{Y})^2}} \tag{1}$$

The correlation coefficient r ranges from -1 to 1, where $r < 0$ indicates negative correlation and $r > 0$ indicates positive correlation. The larger the absolute value of r, the stronger the correlation between X and Y.

For the label ‘y’, there are only two values, "yes" or "no". To streamline the following correlation calculation and modeling process, this article converts the label ‘y’ into a Boolean variable, using 0 to represent "no" and 1 to represent "yes". Fig. 1 shows the correlation heat map between the seven numerical features and the label ‘y’. It can be seen that ‘duration’ has the strongest correlation with ‘y’, with a correlation coefficient of 0.39. The correlations of the remaining features are all low, and only the absolute values of the correlation coefficients of ‘pdays’, ‘previous’, and ‘campaign’ are between 0.05 and 0.10.

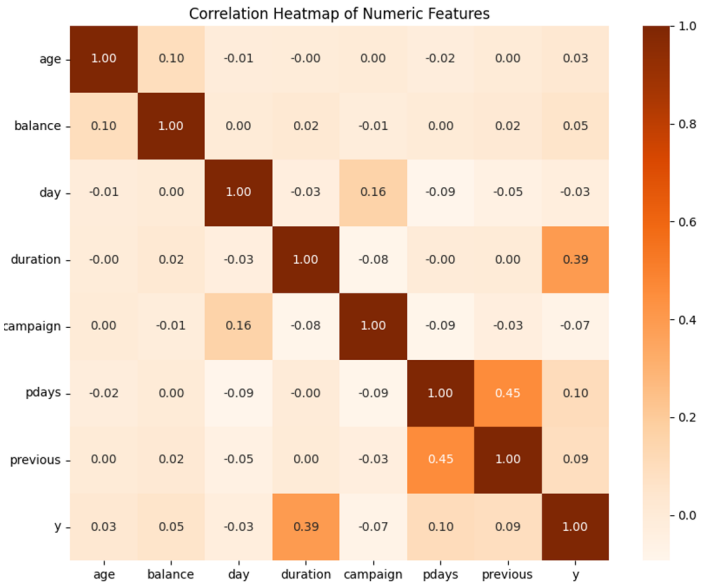

Fig. 1. Correlation coefficient of numeric features (Photo/Picture credit: Original)

The correlation analysis is carried out on the nine categorical features of the training data set. This paper uses Cramér's V coefficient to analyze the correlation between categorical variables. Cramér's V is calculated based on the contingency table, which is a matrix showing the frequency distribution of variables [6]. First, the chi-square statistic χ^2 needs to be calculated, as shown in formula (2):

$$\chi^2 = \sum \frac{(O-E)^2}{E} \tag{2}$$

Where O is the observed frequency; E is the expected frequency, which is equal to the product of the row sum and the column sum divided by the total number of samples. Then, Cramér's V is calculated as shown in formula (3):

$$V = \sqrt{\frac{\chi^2}{n(k-1)}} \tag{3}$$

The value range of Cramér's V is [0, 1], where 0 represents no correlation between variables and 1 indicates a perfect correlation. Fig. 2 shows the correlation heat map between 9 categorical features and the label 'y'. It can be seen that the Cramér's V coefficients of 'poutcome' and 'month' exceed 0.25. Except for the correlation coefficients of 'housing' and 'job', which are above 0.1, the correlations of the remaining features with 'y' are relatively weak.

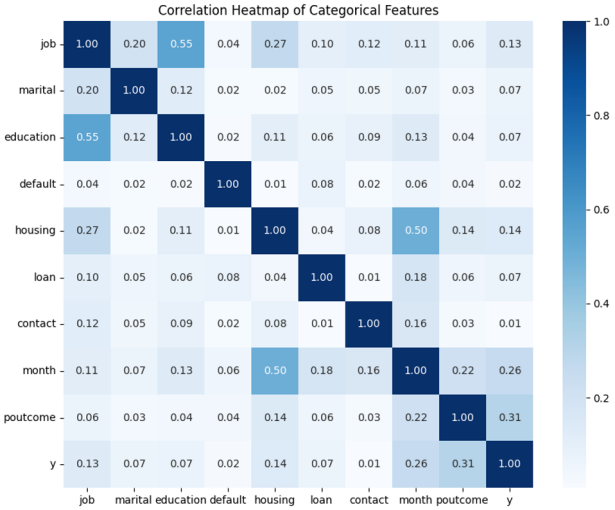

Fig. 2. Correlation coefficient of categorical features (Photo/Picture credit: Original)

Based on the above correlation analysis, this study finally keeps four numerical features, namely ‘duration’, ‘pdays’, ‘previous’, and ‘campaign’, and four categorical features, namely ‘poutcome’, ‘month’, ‘housing’, and ‘job’, for a total of eight features.

In order to incorporate categorical features into the calculation of the model, this paper performs One-Hot encoding on multi-category categorical features so as to convert them into processable binary features [7]. Using One-Hot encoding is equivalent to placing all feature variables in the same space, so that the feature can be regarded as a point, and the Euclidean distance between any two categories is $\sqrt{2}$, which is more reasonable for unordered categorical features.

One-hot encoding converts a feature with n values into n-1 binary features. Taking ‘poutcome’ as an example, this feature originally had four values. After One-Hot encoding, it changes from one variable ‘poutcome’ to three Boolean variables, namely ‘poutcome_other’, ‘poutcome_success’, and ‘poutcome_unknown’. When the original value is "success", the above variables take 0, 1, and 0 respectively. Similarly, similar values can be used to represent "other" and "unknown". When the above three variables are all 0, it means "failure".

After encoding, feature data of 29 dimensions were finally obtained.

Since each numerical feature in the data set has its own measurement unit, and the value ranges of different features vary greatly, all numerical features need to be normalized to eliminate the impact of the dimension on the experiments.

This paper applies the Min-Max normalization approach to map the original value to [0, 1] through linear change [8,9]. The calculation method is shown in formula (4):

$$X^* = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4}$$

After normalization, all numerical features become float types with a value range of [0, 1].

The distribution of whether customers subscribe to term deposits (y) in the processed training data set is statistically analyzed. As shown in Fig. 3, there are 30,545 customers who refuse to subscribe to term deposits, and only 4,009 customers who successfully subscribe.

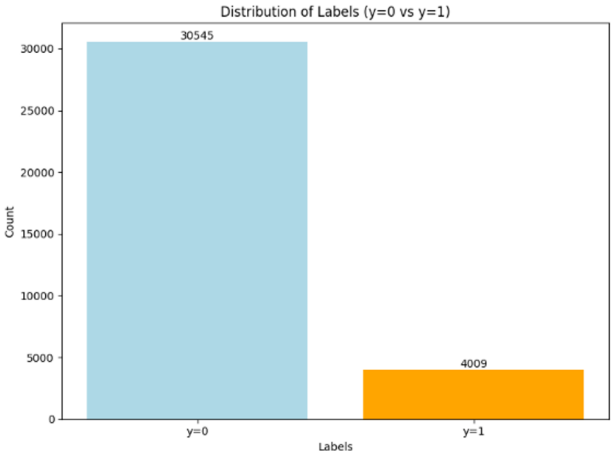

Fig. 3. Statistical chart of whether customers subscribe to fixed deposits (Photo/Picture credit: Original)

Obviously, the positive and negative samples in this data set are extremely unbalanced, so this paper uses the Synthetic Minority Oversampling Technique (SMOTE) oversampling method to cope with this.

The SMOTE enhances the random oversampling algorithm, which merely duplicates existing samples to raise the count of minority class samples. To prevent model overfitting, SMOTE generates new minority class samples through interpolation, rather than simply duplicating the existing ones. The specific process is as follows:

a) Traverse the minority class samples.

b) For each minority class sample, find its nearest K similar samples.

c) Randomly select a sample from these K samples and interpolate between the sample and the original sample to generate a new sample. Its interpolation method is shown in formula (5):

$$x_{new} = x_i + \lambda \cdot (x_{nn} - x_i) \quad (5)$$

Where x_{new} is the newly generated sample, x_i denotes the current minority class sample, x_{nn} is a sample randomly selected from the K nearest neighbor samples,a nd λ is a random number between 0 and 1.

d) Repeat the above steps until the target number of minority class samples is reached.

After Smote sampling, the training data set now includes 61,054 samples, of which positive and negative samples account for half each.

2.3 Methods

For the binary classification task of the bank's marketing forecasting, this paper selected three methods: random forest, KNN algorithm, and logistic regression for modeling and forecasting.

2.3.1 Logistic regression

Logistic regression is a technique for classification in supervised machine learning algorithms, which usually utilizes known features to predict the value of a discrete target variable. The outcome calculated by logistic regression fitting is a probability value (0-100%),

which is mapped to the classification of the final predicted target variable according to the determined classification threshold [10].

The basic idea of logistic regression is applying the logistic function to map the output of linear regression to between (0, 1), as shown in formula (6):

$$P(y = 1|\boldsymbol{X}) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n)}} \quad (6)$$

Where β_0, β_1, $\cdots$, and β_n are model parameters and $\boldsymbol{X}$ is the feature variable.

The logistic regression model employs the maximum likelihood estimation technique to determine parameters that optimize the chance of accurate classification on the training dataset. The loss function used is the cross-entropy loss, as shown in formula (7):

$$Loss = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)] + \lambda \cdot \Omega(\theta) \quad (7)$$

Where the first term on the right side of the equation is the data fitting term, y_i is the true value, $\hat{y}_i$ is the predicted value; the second term is the regularization term, λ is the regularization coefficient, $\Omega(\theta)$ is the regularization function, and θ is the parameter of the model.

2.3.2 Random forest

Random forest is an ensemble learning method based on decision trees. It constructs multiple decision trees and combines them with a voting mechanism to perform classification or regression. Its basic idea is to reduce overfitting by introducing randomness to improve the generalization of the model.

The algorithm flow of the random forest is as follows:

a)Use Bootstrap Aggregation to randomly extract multiple subsample sets from the original training data set with replacement for training multiple decision trees.

b)At each node of each tree, a part of the features is randomly selected for optimal splitting, which reduces the dependency between related trees and enhances the model's capacity for generalization.

c)Each tree is trained based on the randomly extracted samples and features until a complete decision tree is generated, ensuring the low deviation of each tree.

d)For classification problems, the final classification result is determined by majority voting.

2.3.3 KNN

KNN algorithm is an instance-based non-parametric supervised learning algorithm that selects the K nearest neighbor to the input sample based on the distance between the input sample and the training sample and uses the information of these neighbors to make predictions [11]. For classification tasks, KNN usually uses the majority voting method.

The algorithm flow is:

a) For each test point, calculate the distance between it and all samples in the training set.

b) Find the K training samples closest to the test point.

c) Classify the point according to the predominant category among the K nearest neighbors.

2.4 Metrics

This paper evaluates the prediction results of the model based on the confusion matrix. For the binary classification task, the size of the confusion matrix is 2×2, as shown in Table 2:

Table 2. Confusion matrix

Actual \ Predicted	Positive	Negative
True	TP	TN
False	FP	FN

In this experiment, Positive indicates that the customer will make a deposit, while Negative indicates that he or she will not. True/False represents the correctness of the prediction. Accordingly, TP, TN, FP, and FN denote different results.

Based on the confusion matrix, the study can get the following six evaluation indicators for experiments:

Accuracy indicates the proportion of samples that are correctly classified among all samples, as shown in formula (8):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{8}$$

Precision indicates the probability that a sample that is predicted to be positive will actually be positive as well, as shown in formula (9):

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

Recall indicates the probability that the prediction is positive among all actual positive samples, as shown in formula (10):

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

F_1-score is an evaluation index obtained by comprehensively considering the precision and recall. For this imbalance problem, F_1-score reflects the model's ability to recognize the minority class. The F_1-score is the harmonic mean of the precision and recall as shown in formula (11):

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{11}$$

The PR curve is a curve that describes the relationship between precision and recall. When the algorithm classifies a sample, it calculates the confidence of the sample, that is, the probability that the sample is positive. By selecting a suitable threshold, the samples are divided, and those with a confidence greater than the threshold are predicted as positive examples, and vice versa. By adjusting the threshold, different precision and recall rates can be obtained, and a curve can be drawn. It is generally believed that the larger the area under the curve (AUC) the better the model performs.

The ROC curve shows the relationship between the recall rate and false positive rate of the model under different thresholds, where the false positive rate represents the ratio of all negative samples that are incorrectly predicted. Similar to the PR curve, the AUC reflects the performance of the model.

3 Experimental analysis

3.1 Results

Based on this data set, this paper sets up a comparative experiment on the fitting performance of different models under the condition of oversampling or not as shown in Table 3:

Table 3. Comparison of different experimental conditions.

Model	SMOTE	Hyperparameters	Accuracy	Precision	Recall	F_1
Logistic regression	Yes	C=10	84.62%	41.91%	78.43%	54.63%
K-nearest neighbor		n_neighbors=3	82.49%	34.87%	55.69%	42.88%
Random forest		max_depth=None n_estimators=200	88.92%	52.45%	66.18%	58.52%
Logistic regression	No	C=1	84.30%	40.60%	80.18%	53.91%
K-nearest neighbor		n_neighbors=3	88.47%	49.37%	27.70%	35.49%
Random forest		max_depth=10 n_estimators=200	85.18%	42.59%	84.53%	56.64%

The hyperparameter C of logistic regression is the inverse of its regularization coefficient λ; the hyperparameter n_neighbors of KNN indicates the number of selected neighbor points; the hyperparameter max_depth of random forest is the maximum depth of each tree, and n_estimators represents the number of trees. The hyperparameters in Table 3 are relatively well-suited parameters determined by grid search for the current data set. These configurations optimize the performance of the models.

3.1.1 Hyperparameter Analysis

As can be seen from Table 3, when no oversampling is performed, the logistic regression model tends to overfit the majority of class samples, so a smaller C value is required to ensure its regularization strength to prevent overfitting. After SMOTE processing, the C value of logistic regression increases from 1 to 10, indicating that the model regularization strength is reduced, the data balance is improved, and the model can learn the characteristics of minority class samples more fully.

Similarly, when not oversampled, the imbalance of the data set requires random forests to limit its maximum depth to minimize overfitting. After SMOTE oversampling, the balance of the data is improved, and random forests can use deeper trees to capture the complex features in the data fully. With more minority class samples, the model can more confidently build complex decision boundaries.

3.1.2 Performance Analysis

The accuracy of several models under both conditions is above 80%, and some are even close to 90%, but the F_1-score is not ideal. The accuracy rate cannot fully reflect the performance of the model, because as long as the model predicts the majority of negative samples in the test set accurately enough, it can achieve a high overall accuracy rate. According to the F_1-score, the three models have very limited prediction capabilities for a small number of positive samples.

After SMOTE oversampling, the precision of random forest increased by about 10 percentage points, but the recall rate decreased by 18 percentage points. The reasons for this

result may be as follows: First, SMOTE generates numerous positive samples, allowing the model to better learn the features, thus improving the precision rate, but also being overconfident in the characteristics of positive samples and misjudging some positive samples as negative samples; second, SMOTE changes the data distribution, making the decision boundary more complex, which may cause the model to become more conservative in predicting the positive class to reduce the number of false positives, so the precision rate increases, while inevitably missing some positive samples, resulting in a decrease in recall rate.

After SMOTE oversampling, the precision of KNN dropped significantly, while the recall rate increased significantly. The reason may be that KNN is a neighborhood-based algorithm. The introduction of synthetic positive samples, especially when these synthetic samples are close to some negative samples, plus the number of neighbors is only 3, can easily lead to some negative samples being misclassified as positive. Therefore, the FP in the confusion matrix increases and the FN decreases, which in turn increases the recall rate, reduces the precision rate, and also increases the F_1-score to some extent.

The intuitive reflection of the performance of several models under two conditions is shown in Fig. 4 and Fig. 5:

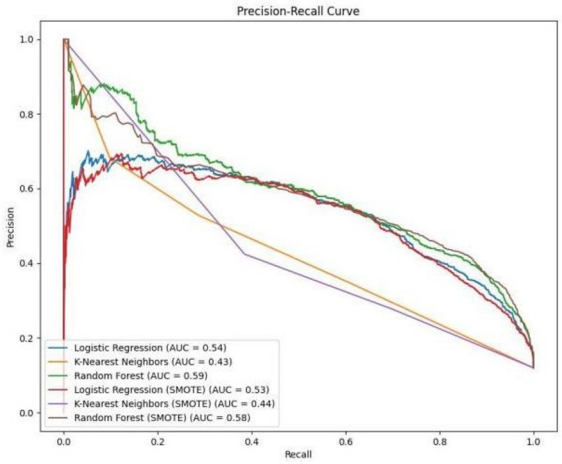

Fig. 4. Precision-Recall Curve (Photo/Picture credit: Original)

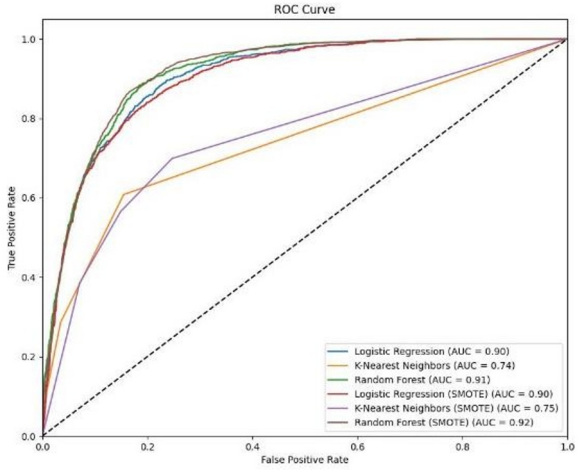

Fig. 5. Receiver Operating Characteristic (ROC) Curve (Photo/Picture credit: Original)

As illustrated in Fig. 4 and Fig. 5, for this data set, regardless of whether SMOTE oversampling is performed, random forest is slightly better than logistic regression and significantly better than the KNN algorithm.

For complex data distribution, random forest, as an integrated method, builds multiple decision trees and performs voting to make predictions, which usually has good noise resistance and generalization capabilities. However, logistic regression is more difficult to handle complex relationships or highly nonlinear data. In the data set of this experiment, only the 'duration' feature has an absolute value of the Pearson correlation coefficient with the label y exceeding 0.1, which also leads to the unsatisfactory decision boundary learned by the logistic regression model that relies too much on linear assumptions.

The KNN algorithm relies on nearby samples for classification and is very sensitive to data noise and imbalance. After generating new samples through SMOTE, it is easy to learn the wrong decision boundary. Therefore, its fitting effect is far inferior to the other two models.

3.2 Limitations and room for improvement

This paper selects features based on correlation analysis. The Pearson coefficient mainly measures the linear correlation between variables. If there is a nonlinear relationship between the feature and the label, the Pearson coefficient cannot capture it. Cramér's V coefficient only measures the correlation between variables but cannot determine the causal relationship, which may make the selected features actually have no direct impact on the label y. Therefore, for the feature engineering of this experiment, the feature selection criteria can be improved, such as adding considerations for nonlinear relationships, using chi-square tests or mutual information methods to evaluate the importance of features, and finally selecting more reasonable and effective features for modeling.

SMOTE oversampling balances the categories by producing new synthetic samples. However, when the distribution of minority-class samples is complex or sparse, these synthetic samples are likely to be affected by noise or unrepresentative samples. The distribution of new samples is inconsistent with the original feature space, which may cause the model to overfit the synthetic samples. To solve this problem, people can consider using other oversampling methods or a combination of oversampling and undersampling to improve.

The three models used in this experiment did not perform well enough in terms of precision, recall, and F_1-score. The grid search applied to adjust the hyperparameters is not efficient and may have missed a better parameter combination. The three models all have their own shortcomings for this dataset. The study can consider model integration methods, combining the prediction results of multiple models, and performing stacking or voting to optimize the prediction performance.

4 Conclusion

Based on the bank telemarketing dataset, this paper uses three methods, logistic regression, KNN algorithm, and random forest, to predict whether customers will subscribe to fixed deposits, and compares the models' performance in detail under different circumstances.

The results show that the random forest model performs better overall, with a higher F_1-score than the other two models. However, due to the limitations of its linear assumptions and insufficient learning of certain nonlinear features, the performance of logistic regression is slightly inferior. The KNN algorithm performs relatively poorly in the experiment due to its sensitivity to noise and unbalanced data. Through comparative analysis, this paper also demonstrates that for unbalanced data, SMOTE oversampling can effectively enhance the

model's recognition ability for minority class samples, but in some cases it may overfit, resulting in a decrease in precision or recall.

This paper provides a reference for data analysis and potential customer positioning of bank telemarketing, which will help the banking industry optimize marketing strategies and reduce operating costs in the fierce market competition. In future research, it is possible to consider introducing more nonlinear feature selection methods, exploring more stable balanced data methods, selecting more complex models, or combining integrated methods to improve prediction performance.

References

1. S. Palaniappan, A. Mustapha, C.F.M. Foozy, R. Atan, Customer profiling using classification approach for bank telemarketing. Int. J. Inform. Visu. 1, 214–217 (2017)
2. A. Lawi, A.A. Velayaty, Z. Zainuddin, On identifying potential direct marketing consumers using adaptive boosted support vector machine, in Proceedings of the 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), Kuta Bali, Indonesia, (2017) 1-4
3. S. Moro, P. Rita, P. Cortez, Bank marketing. UCI Mach. Learn. Repos. (2014)
4. F. Tang, H. Ishwaran, Random forest missing data algorithms. Stat. Anal. Data Min.: The ASA Data Sci. J. 10, 363–377 (2017)
5. P. Sedgwick, Pearson's correlation coefficient. BMJ 345, e4483 (2012)
6. W. Bergsma, A bias-correction for Cramér's V and Tschuprow's T. J. Korean Stat. Soc. 42, 323–328 (2013)
7. J. Brownlee, Why one-hot encode data in machine learning? Machine Learning Mastery (2017). https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/
8. P. J. Muhammad Ali, Investigating the Impact of Min-Max Data Normalization on the Regression Performance of K-Nearest Neighbor with Different Similarity Measurements, ARO 10, 85-91 (2022)
9. H. He, E.A. Garcia, Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21, 1263–1284 (2009)
10. J. Tolles, W.J. Meurer, Logistic regression relating patient characteristics to outcomes. JAMA 316, 533–534 (2016)
11. P. Hall, B.U. Park, R.J. Samworth, Choice of neighbor order in nearest-neighbor classification. Ann. Stat. 36, 2135–2152 (2008)