Classification Analysis On A Bank Marketing Campaign

Group 6

Shutong Fan

Xinyu Yang

Xueting Deng

Yingnan He

04/27/2022

IE 7374

Abstract

Our project is mainly focusing on classification analysis, whether a client will subscribe to a term deposit after a bank marketing campaign through phone contact. The data is related to a direct telemarketing campaign conducted by a Portuguese banking institution. During the phone call, bank agencies collected clients' features, which will be the predictors in this project. After exploring and preprocessing the dataset, we build models to help the bank find those clients who are most likely to purchase the term deposit through telemarketing calls. The performances of all models are evaluated by computing the recall and F1 score of the class of subscription (class1).

According to the performance of F1 score and efficiency, Naive Bayes is the best performing algorithm generally. If the bank wants to recognize clients who are likely to subscribe the deposit without considering the cost, we recommend Neural Networks which has the highest recall rate among all models.

Contents

I. Introduction

In recent years, there are different types of the marketing campaign in the banking market. Telemarketing is one of the most common marketing campaigns in the real world. The dataset we used for our final project is related to a direct telemarketing campaign conducted by a Portuguese banking institution. During this telemarketing campaign, the banking institution collected clients' personal information and customers' previous contact records, as well as social and economic context attributes during the phone call. At last, the bank recorded whether it has successfully sold its term deposit to clients.

Here data mining and machine learning techniques come into play. With a client's information and society context indexes, supervised classification models are trained to predict whether the client is likely to subscribe to a banking product.

There are 4 classic classification models we are going to build this time, Logistic Regression, Naive Bayes Classifier, Support Vector Machines (SVM), and Artificial Neural Networks (ANN). Logistic Regression is a discriminative model that estimates parameters directly from the training data. The naive Bayes approach simplifies the computation process, however, it assumes that the predictors are conditionally independent given the target class. Support Vector Machine tries to find an optimal hyperplane with maximized margin to classify data points. With Kernel Trick, it can deal with non-linearity and high dimensions. Neural Networks, consisting of a network of functions, can learn potential relationships among predictors and the response variable, then analyze new data.

In our project, we plan to compare and analyze the performance of different supervised machine learning models. We propose to provide a good-performing model that can predict the result of telemarketing activity for selling a long-term deposit. To be specific, we should be able to answer the questions in two scenarios in a real business. The question in the first scenario is when a client receives a marketing campaign, what is the probability that he (or she) will buy the deposit. The second one is before the customer receives the campaign, how much effort the bank should input if they want to develop him into a real customer.

Generally, in this project, we have four main contributions:

- Data exploration
- Feature engineering and selection
- Modeling
- Model performance analysis and conclusions

This report will cover these contributions in later sections.

II. Data Description

There are 20 input variables in this dataset, half of them are numeric variables, and the rest are categorical variables, shown in the table below:

Categorical Variables	Numeric Variables
1. Job Type	1. Age
2. Marital Status	2. Last Contact Duration (in seconds)
3. Education Level	3. Number of Contacts After Previous Campaign
4. Credit in Default	4. Number of Contacts Before This Campaign
5. Housing Loan	5. Employment Variation Rate
6. Personal Loan	6. The Consumer Price Index
7. Contact Communication Type	7. Consumer Confidence Index
8. Last Contact Month of Year	8. Euribor* 3-month Rate
9. Last Contact Day of Week	9. Passing Days After Last Campaign
10. Previous Marking Campaign Outcome	10. Number of Employees

** The Euro Interbank Offered Rate (Euribor) is a daily reference rate, published by the European Money Markets Institute (Source: Wikipedia).*

The response is a binary unsuccessful or successful subscription of a long-term deposit. In 41188 records, 4640 (11.3%) records are related to success. Thus, this data set is imbalanced. We shall use oversampling or undersampling techniques to handle imbalanced data.

III. Explanatory Data Analysis (EDA)

Dropping Missing Values

In the original dataset, we have 41188 records with 20 predictors. However, values in some cells are "unknown". In other words, they are missing values. The table below shows the number of missing values in corresponding predictors.

Predictor	Missing Values (Unknown)
Job	330
Marital	80
Education	1731
Default	8957
Housing	990
Loan	990

We replace the "unknown" with none, then drop these missing values. This time there are 30488 records remaining for training with a similar previous balance of classes in the response variable.

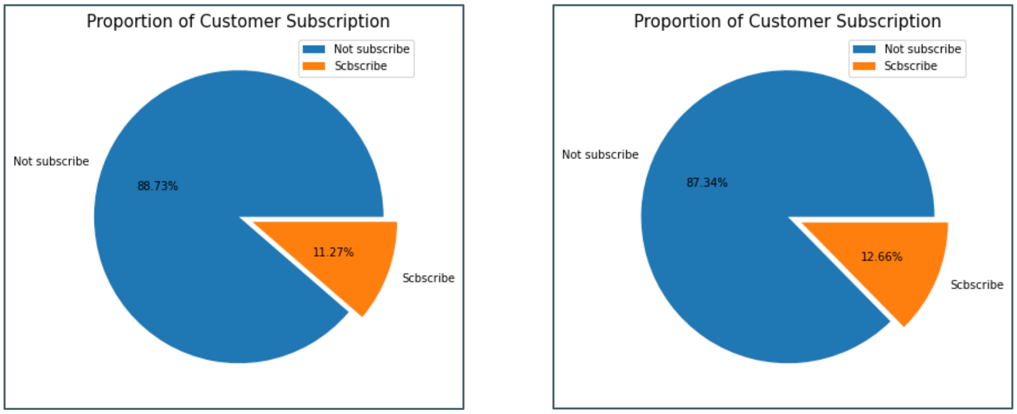

Fig 3.1 Left: before dropping missing values; Right: after dropping missing values.

Checking Class Balance in Response Variable

The y value in this dataset is regarded as after a phone call of selling, whether a client will subscribe to a long-term deposit or not, where "yes" means success, "no" means failure. The distribution of two results is shown in the chart below.

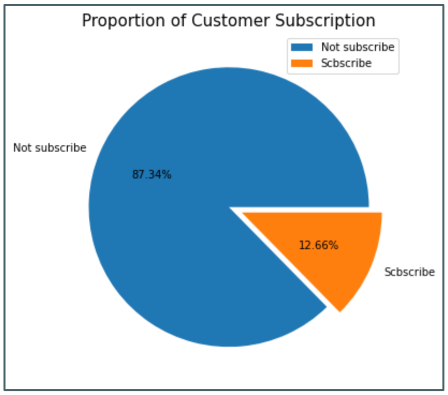

Fig 3.2 Response Class Distribution

According to the chart, we can see that the records in 2 classes are not evenly distributed, since the number of clients who did not subscribe to the deposit is much larger. Therefore, we will train models with oversampling or undersampling method on the rare class to guarantee the predictive performance of models.

EDA Of Categorical Predictors

We have 10 categorical predictors in this dataset. Before feature selection, we first made bar plots for each feature of the categorical predictors grouped by the binary contact outcome.

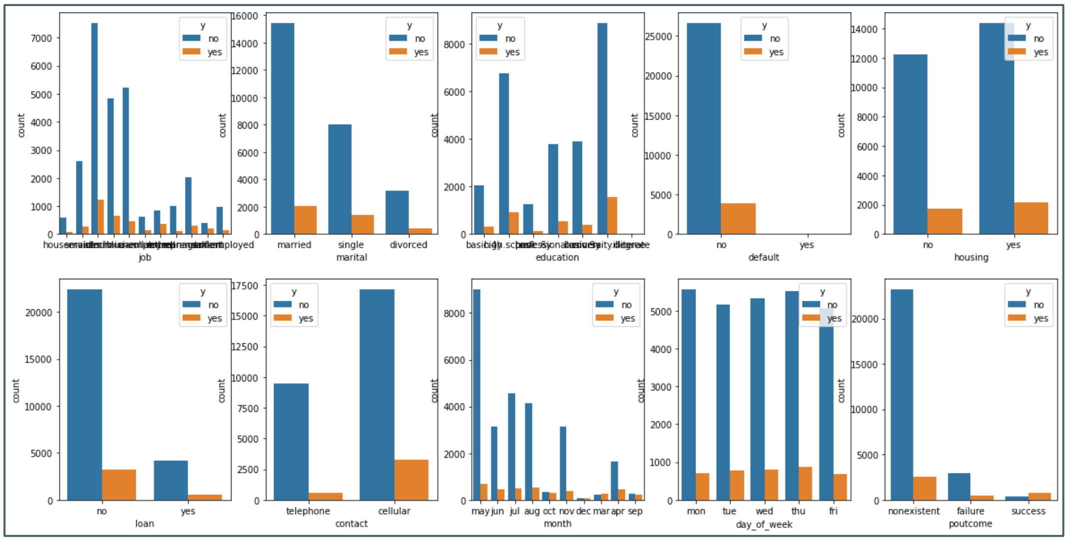

Fig 3.3 Categorical Variable Distribution by Response Class

Even though it is difficult to verify if there is a significant difference in contact outcome in different features of one categorical variable, we generally have a vision that how do these variables distribute in 2 response groups.

Statistical Testing

We conducted statistical testing to determine whether there are statistically significant differences between the predictors and the response variable.

Since both predictors and response variables are categorical, we choose chi-square test. The chi-square testing result of categorical variables is shown in the table below.

Variable	Ho*	P-value
Job	Reject	2.0e-150
Marital	Reject	1.5e-12
Edu	Reject	1.4e-22
Default	Accept	0.83
Housing	Accept	0.08
Loan	Accept	0.38
Contact	Reject	4.90e-139
Month	Reject	0
Day of W	Reject	3.7e-5
P-outcome	Reject	0

**Ho: predictor and response variable are independent.*

EDA Of Numeric Predictors

To better visualize the distribution of numerical predictors in 2 response classes, we make side-by-side box plots for exploring the subscription outcome by different numerical predictors.

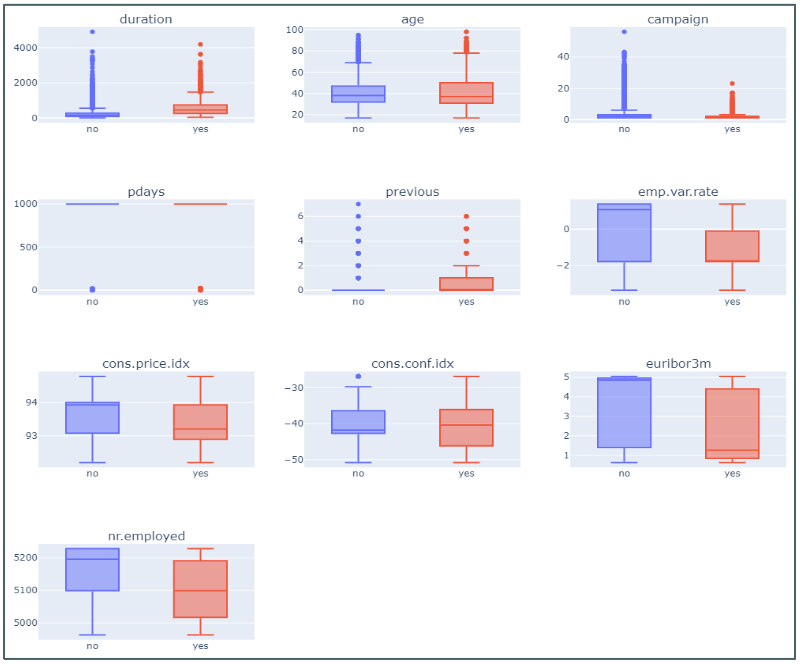

Fig 3.4 Numeric Variable Distribution by Response Class

Statistical Testing

Since the predictors are numeric and responses are categorical, we apply point biserial test here. The point-biserial testing result of numerical variables is shown in the table below.

Variable	Correlation	P-value
Age	0.04873	1.70e-17
Duration	0.39353	0.0
Campaign	-0.06921	1.08e-33
Pdays	-0.32751	0.0
Previous	0.22800	0.0
Emp.var.rate	-0.30536	0.0
Cons.price.idx	-0.12875	7.76e-113
Cons.conf.idx	0.06164	4.65e-27
Euribor3m	-0.31587	0.0
Nr.employed	-0.36423	0.0

Correlation Coefficients

We made a heatmap of correlation coefficients of 10 numerical predictors. Apparently, there is redundancy among attributes of social and economic context, since their coefficients are very high. Thus, we think there can be a linear subspace existing in the original space of 5 dimensions. In later session of feature selection, we will adopt Principal Component Analysis to find such a linear subspace.

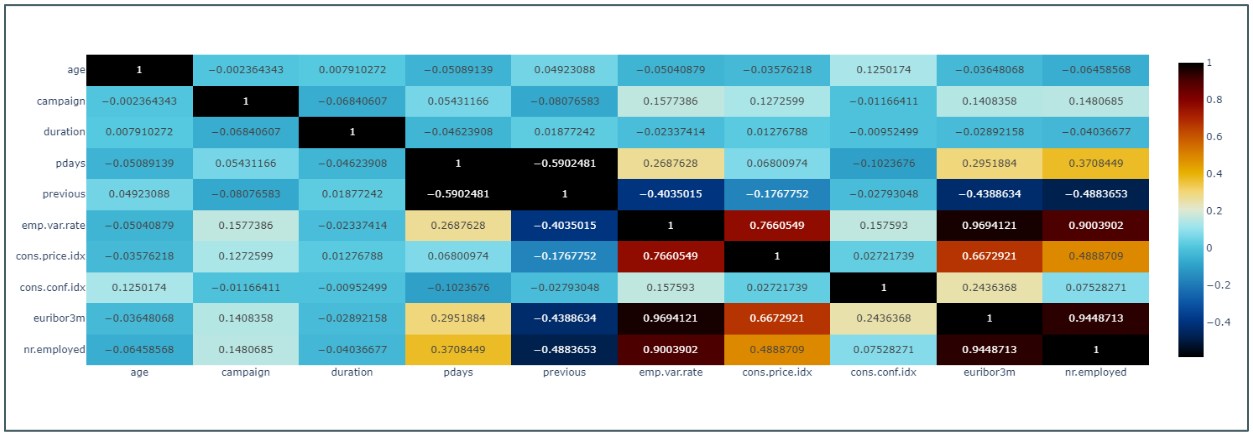

Fig 3.5 Correlation Coefficient Heatmap of Numeric Variables

IV. Feature Selection

Drop variables according to statistical testing result

According to the statistical testing result, we can tell that features in 3 predictors, Default, Housing, and Loan do not significantly influence the frequency of classes of response. Therefore, we directly drop these 3 predictors and keep the other 7 predictors for later model training.

Principal Component Analysis

In the last session, we found that the correlation coefficients between each two social and economics indexes are high, we thought there must be a linear subspace for these numeric variables. Below is the result of the Principal Component Analysis for 5 variables:

1. emp.var.rate: employment variation rate - quarterly indicator (numeric)
2. cons.price.idx: consumer price index - monthly indicator (numeric)
3. cons.conf.idx: consumer confidence index - monthly indicator (numeric)
4. euribor3m: Euribor 3 month rate - daily indicator (numeric)
5. nr.employed: number of employees - quarterly indicator (numeric)

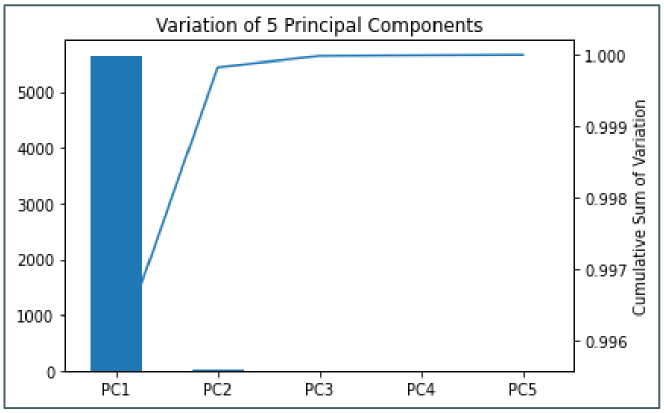

Fig 4.1 Variation Proportion of Principle Components

Based on the bar plot above, the first component captures the most variation of these 5 attributes. Therefore, we will directly replace the original attributes with the first component for later modeling.

Convert numerical predictors to categorical predictors

According to the side-by-side boxplots above, the distribution of pdays (number of days that passed by after the client was last contacted from a previous campaign, where 999 means the client was not previously contacted) is somehow strange. We can see that most clients are not previously contacted, in the other words, those outliers are showing that few clients have been contacted from a previous campaign. Therefore, we convert this numerical variable into a categorical variable “pdays_cat”, where 0 represents the client was not contacted before, and 1 represents there are contact records.

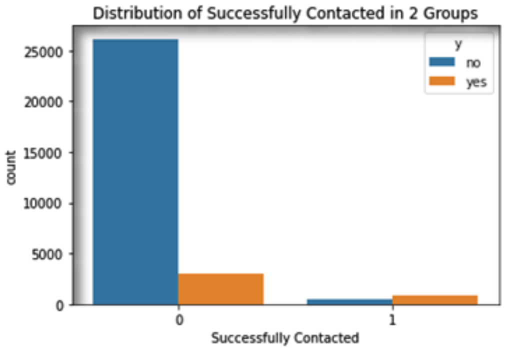

Fig 4.2 Distribution of pdays_cat by Response Class

Moreover, there is another variable “previous”, which is also showing the previous relationship between the bank and clients. Originally, “previous” represents the number of contacts performed before this campaign and for this client. Based on our cross-check, in the group of ones who were not contacted before (pdays_cat = 0), the “previous” values of some records are not 0. We think these clients in such a condition probably were not involved in a previous campaign, but their names were on the contact list of the bank. Thus, combining “pdays_cat” and “previous”, we create a new categorical variable called “pdays_previous”, representing a general contact record with a client. The value of 0 represents the clients who were neither contacted in a previous campaign nor on the bank’s contact list. The value of 1 represents they were not contacted in a previous campaign while they were called by the bank before, showing that they have relationships with the bank to some extent. Last but not least, the value of 2 represents the clients who have the strongest relationship with the bank.

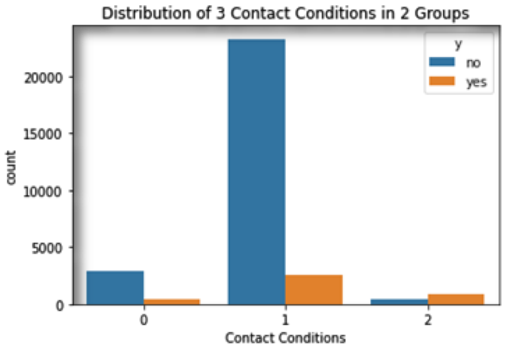

Fig 4.3 Distribution of pdays_previous by Response Class

Obviously, according to the barplot above, people in group 2 are most likely to purchase the term deposit since the orange bar is higher than the blue one.

Profiling by Random Forest

Random Forest is used to measure the remained feature importance in this algorithm. We made a horizontal bar plot showing the features' importance.

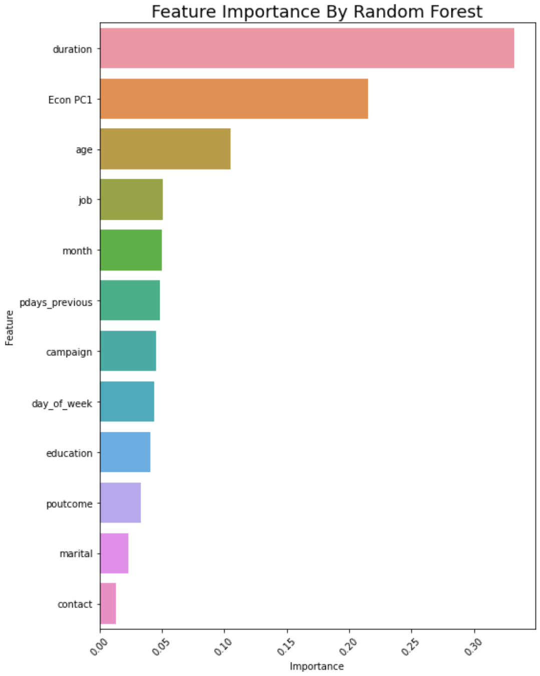

Fig 4.4 Feature Importance by Random Forest

According to this bar plot above, we can see that duration, the first principal component, and age are very important in splitting among a random subset of predictors. This inspired us that duration and PC1 are strong measures for later modeling and analysis.

V. Methods

Model 1 Logistic Regression:

Logistic Regression is considered one of the fastest and cheapest classification models. Rather than using y directly as the outcome variable, we use a function, which is called the logit function.

$$P = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p)}}$$

The logit function can be modeled as a linear function of the predictors. Once the logit has been predicted, it can be mapped back to a probability. Also, the probability of logit belonging to [0, 1]. Since logistic regression has no closed-form formula, we use gradient descent to get the final result. Furthermore, to reduce the impact of overfitting on the model, the penalty is applied in the gradient.

Model 2 Naive Bayes Classifier

Naive Bayes is a generative model that uses the prior and likelihood of data into Bayesian theorem. To avoid having no records restricted to match the new data as well as reduce the amount of data, Naive Bayes assumes the predictors are conditionally independent given the response class. Under this circumstance, we only care about one predictor every time.

$$P(Y = C_i|X_i) = \frac{P(X_1|Y = C_i) \times P(X_2|Y = C_i) \times \ldots \times P(X_i|Y = C_i) \times P(Y = C_i)}{P(X_i)}$$

It provides an accurate class of output but inaccurate propensity compared with the exact Bayesian. We have a binary response in the dataset, the class with a higher propensity score will be a predicted class as a final result.

Data Preparation for Naive Bayes

Naive Bayes relies on frequency to calculate probabilities. After the general data processing, we still have numeric predictors, which cannot be counted. Before modeling the Naive Bayes classifier, we bin 3 numerical attributes, age, phone call duration, and the number of contacts performed during this campaign into ranges of values, with an almost equal number of records in every bin. To verify we have effectively binned the strongest measure, for example, the variable of duration, we plot a line chart showing the percentage of class1 in 6 bins.

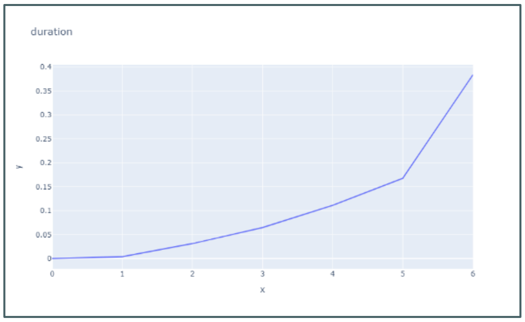

Fig. 5.1 Statistical information of predictors

Here we can see the frequency of class1 from the first bin to the last is increasing. Therefore, the conversion of numeric variables can effectively influence the posterior probability during prediction.

Model 3 Support Vector Machine

The goal of the Support Vector Machine (SVM) is to find a linear hyperplane that can classify correctly and maximize the distance between two categories. We can find only one optimal hyperplane at last. Generally, we proceeded with 3 efforts in Soft Margin SVM modeling.

Through Solving the Primal Problem

We start the SVM model by solving its primal problem, and the cost function is based on Hinge Loss.

$$L = \max(0,1 - y^{i}(x^{i} - b))$$

Combined with the SVM's original objective function, the cost function is written as below.

$$L(w) = \sum_{i=1} \underbrace{max(0, 1 - y_i[w^T x_i + \ b}_{\text{Loss function}}]) + \underbrace{\lambda||w||_2^2}_{\text{regularization}}$$

As we take the gradient of cost function regarding w and b, we can update w and b values by gradient descent. In later analysis, this algorithm is called "SVM_hinge".

Through Solving the Dual Problem

$$\begin{aligned} &\underset{\alpha}{\text{minimize}} && \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha^{(i)}\alpha^{(j)}y^{(i)}t^{(j)}(\mathbf{x}^{(i)})^T\mathbf{x}^{(j)} - \sum_{i=1}^{m}\alpha^{(i)} \\ &\text{subject to} && \begin{cases} \sum_{i=1}^{m}\alpha^{(i)}y^{(i)} = 0 & \text{for } i = 1, \dots, m \\ 0 \le \alpha^{(i)} \le C & \text{for } i = 1, \dots, m \end{cases} \end{aligned}$$

Another method to find an optimal hyperplane for classification in the SVM model is to solve the dual problem. We use the objection scipy.optimize.minimize to minimize the objective function with multiple variables, the alphas.

In later analysis, this algorithm is called "SVM_dual".

Sequential Minimal Optimization with RBF Kernel

Since we have a relatively big dataset, algorithms in soft margin SVM are too inefficient to use for finding α* one by one. Therefore, we try Sequential Minimal Optimization (Platt, 1998), a fast

algorithm for training Support Vector Machines after the two methods mentioned above. In this method, we also introduce Kernel Trick (Gaussian Kernel) to fit the dataset.
In later analysis, this algorithm is called “SVM_kernel”.

Model 4 Neural Network

Neural network is a flexible data-driven (albeit blackbox) method that can be used for classification, prediction, and is the basis for deep learning—a powerful technique that lies behind many artificial intelligence applications. Neural network puts many neurons together in each layer in order to transmit information to each other. And then according to the backpropagation algorithm, fix mistakes and update weights and bias each iteration.
Since the response of the data is binary, for the cost function selection, instead of the softmax method, we use two methods.

- First one is defined by ourselves. We define the first cost function as follows:

$$\mathrm{Loss} = -\frac{1}{\text{output size}} \sum_{\mathrm{i}}^{\text{output size}} \mathrm{y_i}\log(\mathrm{sigmoidy_i^{hat}}) + (1 - \mathrm{y_i})\log(\mathrm{sigmoid}(1 - \mathrm{sigmoidy_i^{hat}}))$$

  And avoiding boundary overflow, we set the lower bound and upper bound of logit value. In addition, the penalty is also added in the process of calculating the gradients and updating the weights with the purpose of avoiding overfitting.

- Second one is sigmoid cross-entropy of TensorFlow.

Oversampling and Undersampling Technique

Since the original distribution of 2 classes in the response variable is not balanced, therefore, there may be not enough information for classifiers to learn patterns to distinguish between the classes. We plan to adopt oversampling (for Logistic Regression, Artificial Neural Networks) or undersampling (Support Vector Machine) techniques for the training dataset to address the imbalance issue. We will keep the original ratio of two classes in the validation dataset.

Data Scaling

Since some algorithms (e.g. SVM) are sensitive to the range of data values, we do feature scaling to standardize the range of features as the last step of general data cleaning and processing.
We use min-max scaling to transform the data so that the features are within a range.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

After scaling, the scaled x is a normalized value.

VI. Results

Performance evaluation

In most instances, the data miners may use accuracy, F1 score, and so on, to evaluate the model performance. However, the most important thing in the project is which class is the class of interest. In bank marketing classification, the customers who are likely to subscribe to the product are the interest of class (in this dataset, response = 1). Because the purpose of the bank is to recognize these bunch of users and have some strategies on them to convert to actual customers. If they can identify potential customers more accurately, it will save a lot of money and time, and convert potential customers to actual customers more efficiently. Therefore, we put much attention on the performance of class 1 rather than blindly pursue the effect of the overall model.

Model 1 Logistic Regression

First of all, Logistic regression is applied on the original data. As we can see from the table and confusion matrix, the performance of class 0(Not subscript) is nice here: recall value achieved 0.98 and f1 score is 0.94. However, our class of interest is class 1(Subscript), which has only 0.25 recall value. That means that the model only can recognize 25% subscript customers successfully from actual subscript customers. Therefore, we try to oversample the training data and keep the balance of the two classes.

Tab. 6.1.1 Model performance of Logistic Regression (Non-Oversampling)

	Precision	**Recall**	**F1 score**
0(Not Subscript)	0.90	0.98	0.94
1(Subscript)	0.7	0.25	0.37

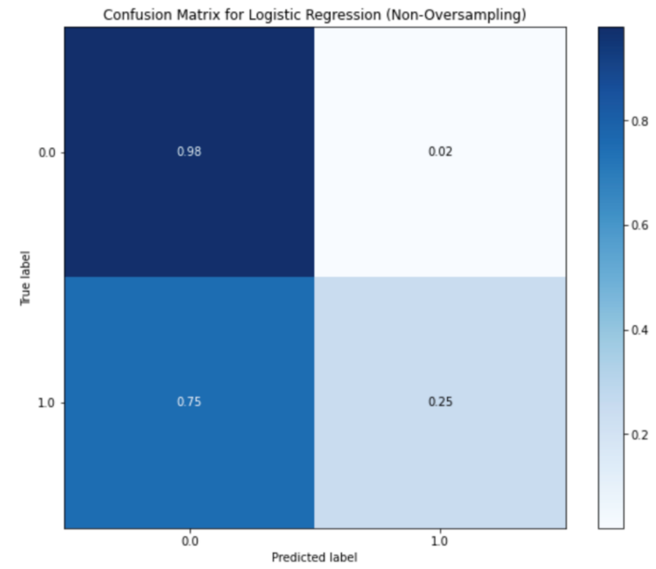

Fig. 6.1 Confusion matrix for Logistic Regression (Non-oversampling)

After oversampling, the recall value of the class of interest attains 0.83, which is a huge promotion compared with the non-oversampling model (recall of class 1: 0.25). Even with the increasing recall of class 1, the precision of class 1 sacrifices at the same time. Also, The class 1 does not perform too badly after oversampling.

Tab. 6.1 Model performance of Logistic Regression (Oversampling)

	Precision	**Recall**	**F1 score**
0(Not Subscript)	0.97	0.84	0.90
1(Subscript)	0.43	0.83	0.57

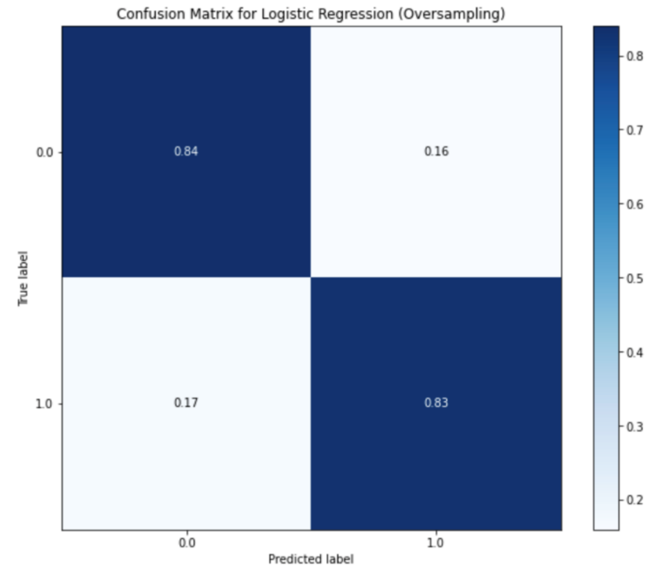

Fig. 6.2 Confusion matrix for Logistic Regression (Oversampling)

With the purpose of improving the model performance, or in other words improving the recall value of class 1. Another method we tried is tuning the cut-off value of the model. Once the threshold is reduced, the probability of being predicted to be class 1 is higher (previously we used 0.5 as default value). To find the optimal cut-off value, we draw the line plot as follows. The x axis is the cut-off value, the orange line is the probability of recall of class 1 and the blue line is the probability of accuracy of the total model. Our goal is to find the cut-off value that maximizes the recall of class 1 and accuracy. So the optimal threshold is found at the intersection of two lines, which is 0.49. Then we apply the optimal cut-off value into the model to compare the performance.

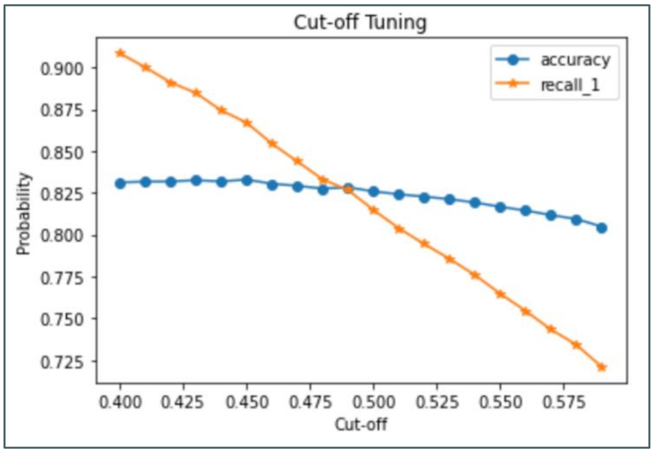

Fig. 6.3 Cut-off tuning

Even though the advancement is not too large, the bank can tune the cut-off value optionally according to their needs. If they think the recall is more important than accuracy, they can continue reducing the cut-off value.

Tab. 6.2 Model performance of Logistic Regression (cut-off = 0.49)

	Precision	**Recall**	**F1 score**
0(Not Subscript)	0.97	0.84	0.90
1(Subscript)	0.43	0.84	0.56

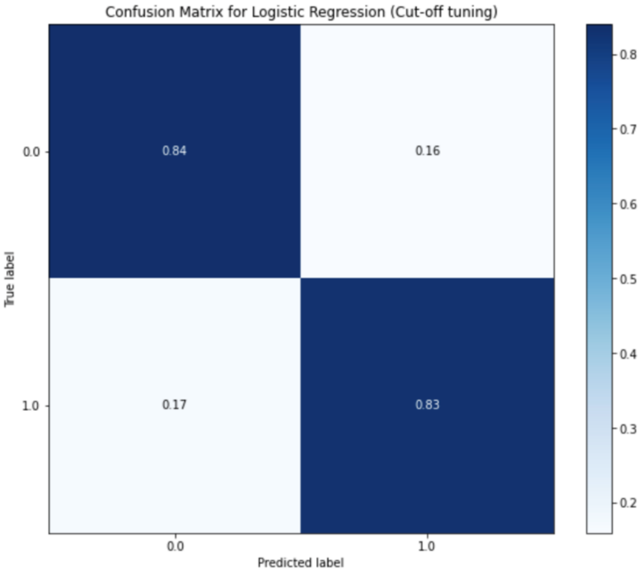

Fig. 6.4 Confusion matrix for Logistic Regression (cut-off = 0.49)

Model 2 Naive Bayes

In the Naive Bayes classifier, we did not train the model on the oversampled dataset because the prior and probability will change after oversampling the data. Moreover, the data will not follow the real-world scenario since only a few people will subscribe to a product from a bank after the campaign.

Tab. 6.3 Model performance of Naive Bayes

	Precision	**Recall**	**F1 score**
0(Not Subscript)	0.92	0.97	0.94
1(Subscript)	0.91	0.74	0.82

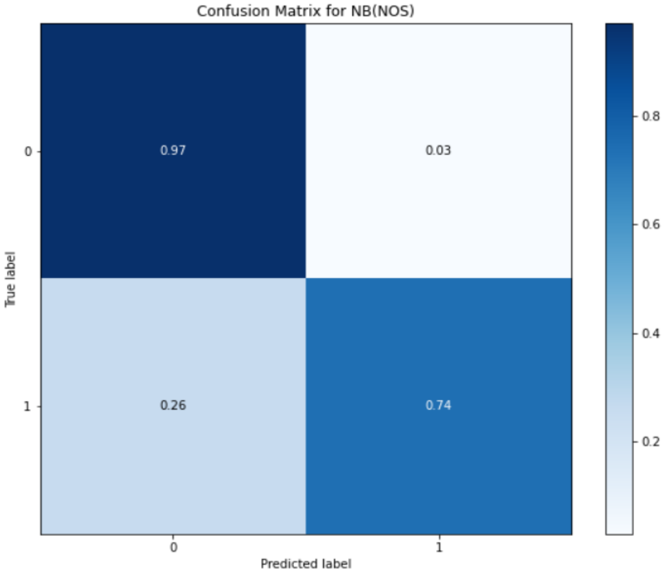

Fig. 6.5 Confusion matrix for Naive Bayes

As we can see from the confusion matrix above, this classifier has a relatively better performance over the other models trained on the original dataset. However, it does not have a strong ability to distinguish the people that will subscribe and the reason we can come up with is the distribution of predictors in 2 classes that overlap with each other.

To optimal the performance, we loop through from 2-20 to find out the best number of bins for each predictor. In addition, we combined two predictors 'pdays' and 'previous' since this can help the model easily pick out the people that have no interest in the campaigns.

There is nothing much we could do to improve the Naive Bayes model and the performance is acceptable, so we just stopped here.

Model 3 Support Vector Machine

SVM_hinge

We applied the soft-margin SVM based on the hinge loss function on the original dataset. The overall accuracy of the whole dataset is 0.88. However, according to the table, the performance of target 0 is very good with the recall of 0.98, precision of 0.90, F1 score of 0.94, while the performance of target 1, the class of interest, is poor, with the recall of 0.21, precision of 0.66, F1 score of 0.32.

Tab. 6.4 Model performance of SVM_hinge (Non-Undersampling)

	Precision	**Recall**	**F1 score**
0(Not Subscript)	0.90	0.98	0.94
1(Subscript)	0.66	0.21	0.32

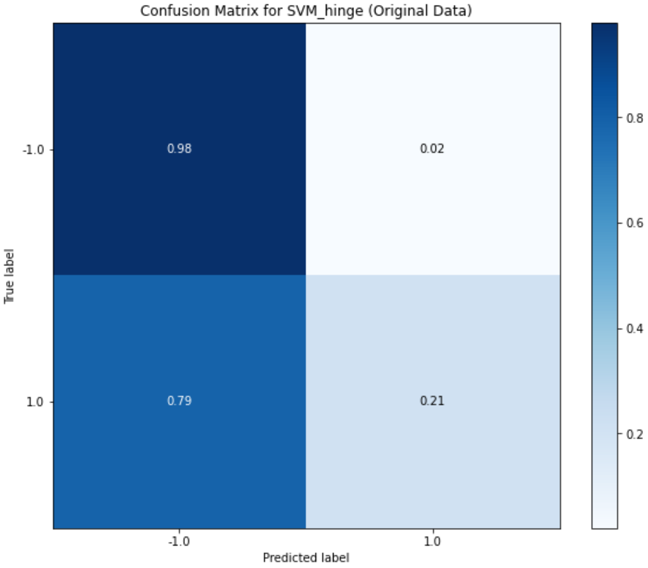

Fig. 6.6 SVM_hinge, learning rate = 0.0001, max iteration = 1000

One of reasons that the model accuracy is high while recall is low is that the data distribution is imbalanced. Therefore, it is necessary to resample class 1 for the dataset. After resampling class 1, there are 3097 class1 and 3097 class0.

We reapplied the SVM_hinge model on the dataset with balanced distribution of class 0 and 1. After undersampling, the recall value of the class of interest can reach 0.86. The recall of class1 has been greatly improved, indicating that the model to classify class 1 has been well trained after undersampling, and the performance on the recognition of both the class1 and class0 is not too bad.

Tab. 6.5 Model performance of SVM_hinge (Undersampling)

	Precision	**Recall**	**F1 score**
0(Not Subscript)	0.97	0.69	0.81
1(Subscript)	0.28	0.86	0.42

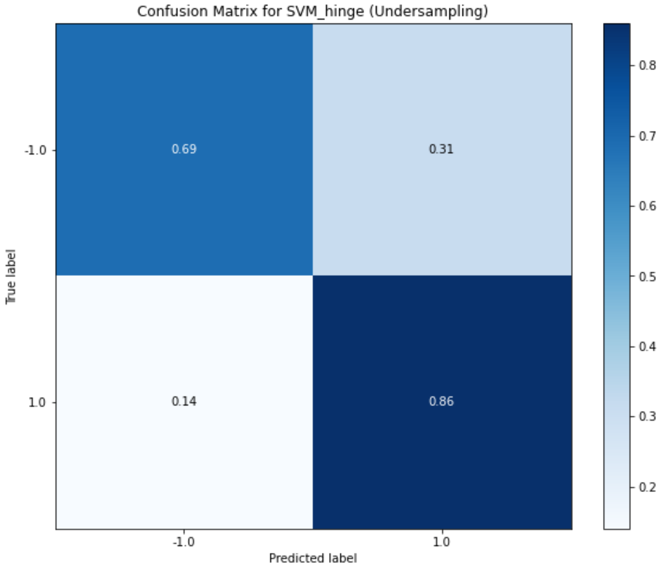

Fig. 6.7 SVM_hinge, undersampling, learning rate = 0.0001, max iteration = 1000

Although the performance of the SVM_hinge model is good, the time-consuming process is a little long. It took 8 minutes to train the training dataset. Compared to the previous Logistic Regression model, the recall of interest class 1 is a bit lower, on the other hand, the recall is higher than Naive Bayes model. In general, SVM_hinge is a very moderate model.

SVM_dual

Similarly, we also need to undersample the data when applying the soft-margin SVM model. Soft-margin SVM model needs to continuously find a hyperplane that can optimally distinguish between data points until the calculated results converge. When there is no obvious linear relationship between the data, it is difficult for a soft-margin SVM model to find the connection between data points. When the kernel function selected by soft-margin SVM is a linear function to distinguish this pile of data, soft-margin SVM needs to be calculated many times to find the convergent hyperplane. That's why the training time of the soft-margin SVM model is extremely long.

In actual training, with just 600 data, the soft-margin SVM model needs to spend around 3 minutes to train. With 6000 data points, it took over 7 hours to run the soft-margin SVM model on a workstation to compute the hyperplane. Due to the algorithm itself and the capabilities of our devices, we do not fit the SVM_dual algorithm with the original data (30488 records) this time.

Tab. 6.6 Model performance of SVM_dual (Undersampling)

	Precision	Recall	F1 score
0(Not Subscript)	0.59	0.47	0.52
1(Subscript)	0.56	0.67	0.61

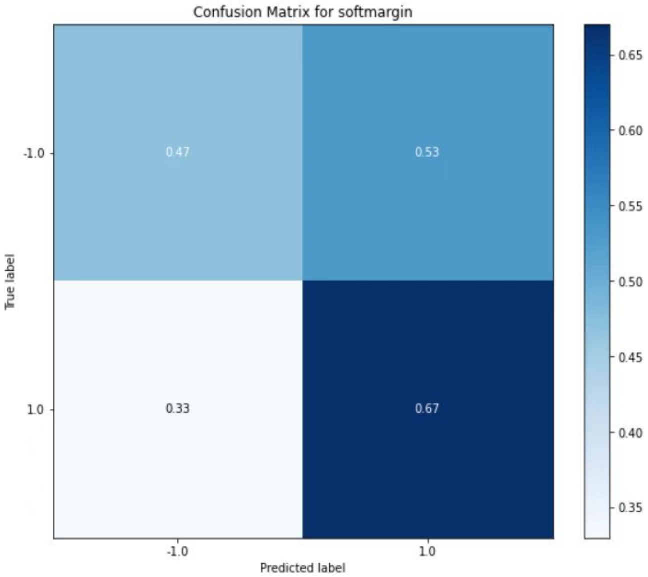

Fig.6.8 SVM_dual, undersampling, learning rate = 0.0001, max iteration = 1000

SVM_kernel

This time, we use the Sequential Minimal Optimization algorithm to accelerate the computation of alpha values. Also, we introduce the Gaussian Kernel trick in the SVM_kernel method. Based on the result, we can see the model increases the recall rate of class1 while sacrificing the precision overall. But generally, the F1 scores have been improved.

Tab. 6.7 Model performance of SVM_kernel (Undersampling)

	Precision	Recall	F1 score
0(Not Subscript)	0.97	0.70	0.81
1(Subscript)	0.29	0.87	0.44

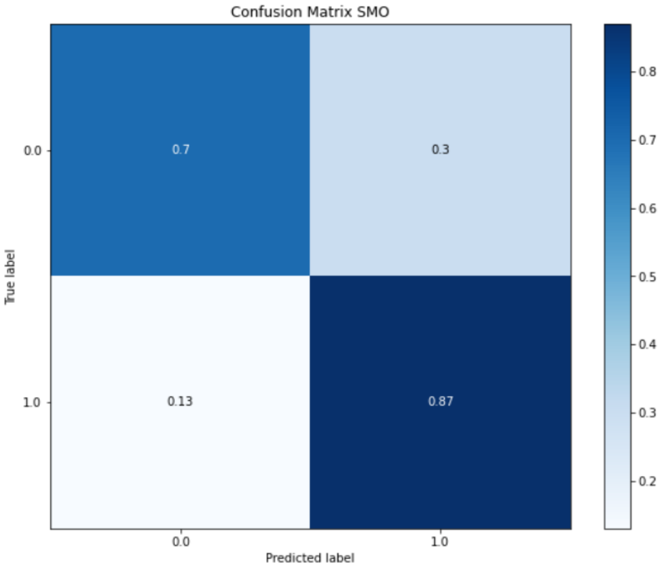

Fig.6.9 SVM_kernel, undersampling, C=1, Kernel gamma = 5

After introducing the Kernel trick, we can see the model improve the F1 score of class1. Therefore, the kernel trick can distinguish class1 and class0 more effectively compared to SVM_hinge and SVM_dual.

Model 4 Neural Network

For some important parameters, like layers, nodes and activation function and so on, we try several combinations and finally choose 2 hidden layers and the first hidden layer has 6 nodes, the second hidden layer has 8 nodes. Because of the binary output, we set the output layer 1 node. For the first layer, we choose relu as an activation function for hidden layer 1 and sigmoid function for hidden layer 2.

- 1st cost function(self-definition)

Overall, the recall value 0.92 of class of interest is the highest one among all models, but sacrificing too much precision, so the final F1 score is not perfect.

Tab. 6.6 Model performance of Neural Network (1st cost function)

	Precision	**Recall**	**F1 score**
0(Not Subscript)	0.98	0.75	0.85
1(Subscript)	0.34	0.92	0.5

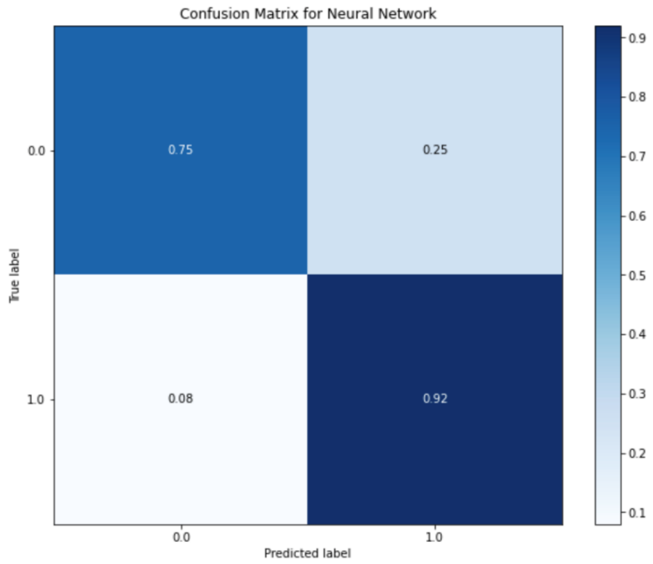

Fig.6.10 Confusion matrix for Neural Network (1st cost function)

- Second cost function (sigmoid cross-entropy)

From the result, we can conclude that this model performs better in both class 0 and class 1 than the previous one, even though the recall value 0.9 is a little lower (compared with 0.92). Besides, the second one has a shorter run time, reducing the time cost. Moreover, the overall model effect is better than others only from the performance perspective.

Tab. 6.7 Model performance of Neural Network (2nd cost function)

	Precision	**Recall**	**F1 score**
0(Not Subscript)	0.98	0.82	0.89
1(Subscript)	0.42	0.9	0.57

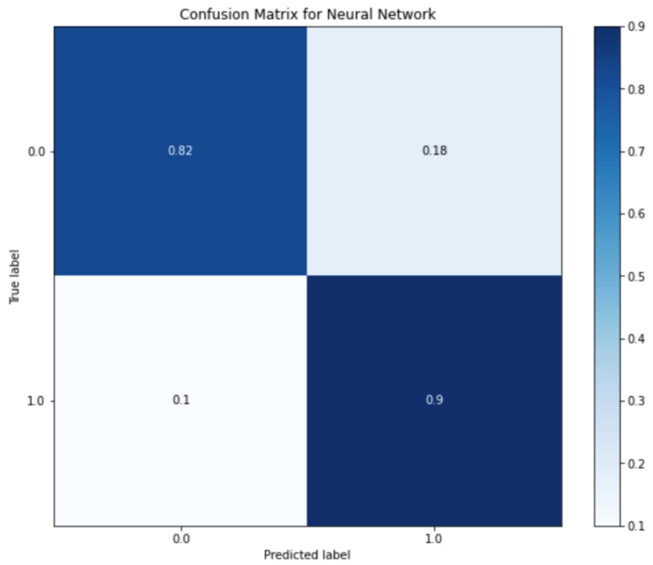

Fig.6.11 Confusion matrix for Neural Network (2st cost function)

VII. Discussion

Advantages and disadvantages

Finally, some discussions are listed as follows to show some advantages and some shortcomings we can optimize later.

Algorithm	Advantages	Disadvantages
Logistic Regression	• Fast in training process with relatively good performance • Easy to interpret	• Vulnerable to overfit
Naive Bayes	• Fast in training process with good performance • No hyperparameters needed	• Certain assumptions on predictors within classes
SVM	• Flexible with introducing kernel trick	• Slow in training • Need time for searching hyperparameters
Neural Network	• Perfect performance	• Slow in training • Easy to overfitting • Hard to interpret

Models Comparison

1. With consideration of the different costs and staff needs of campaigns, we would like to prefer recall of class 1 as evaluation of models in the situation when the expense of phone calls and staff is not expensive, or it is more important and cheaper if the bank gets a new client successfully than loses a customer. That means that the bank wants to improve the rate of correct classification from actual classification 1(recall).

   From the plot, the Neural Network method that has oversampled performs the best in recall respect, though others also perform perfect in recall.
2. Similarly, if the expense of campaigns is expensive, the bank needs to consider their ratio of classify class 1successfully from predicted classification 1(precision).

   Also, the Naive Bayes has the highest precision and others perform not good due to oversample.
3. Under normal circumstances, data miners can also evaluate models by F1 score.

   According to the barplot among all models, we can conclude the blue bar is F1 score and Naive Bayes algorithm has the highest F1 score from all of them. As all know, F1 score is based on both precision and recall value, since other models have low precision, so they do not behave very well in total.

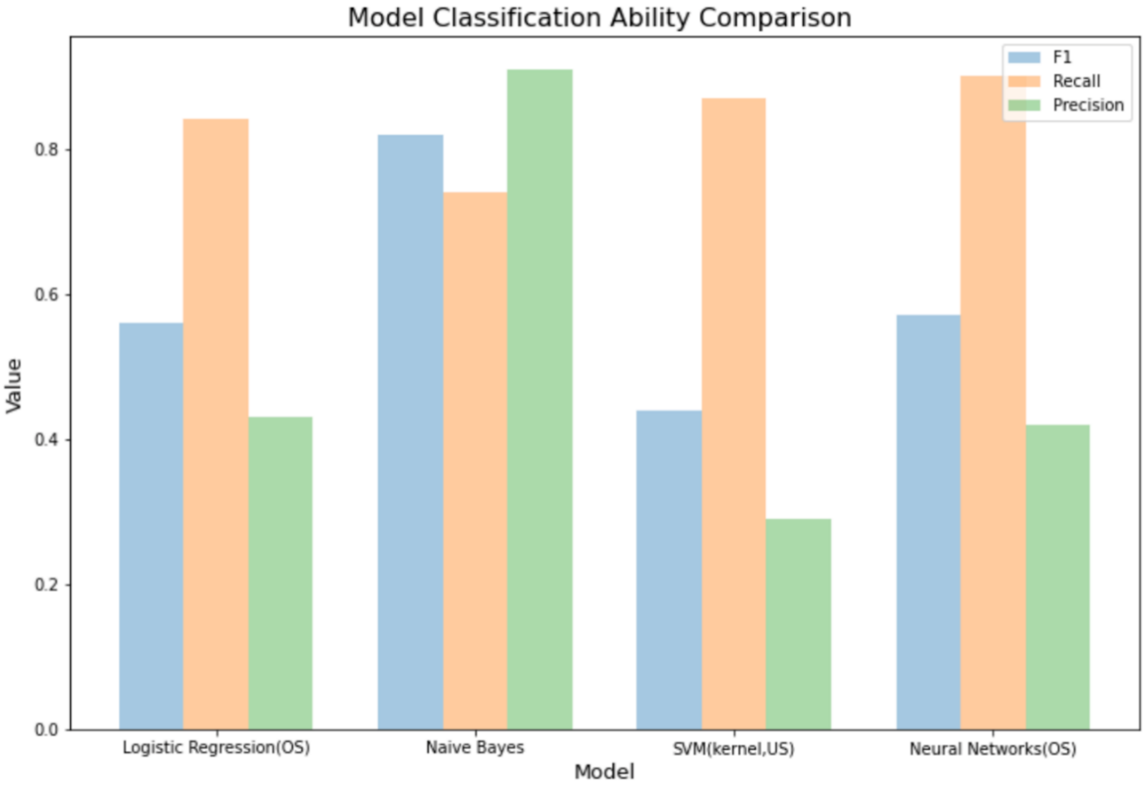

Fig. 7.1 Model classification comparison

Improvement

Data Encode

Now we encode all predictors which we think should encode by using basic knowledge. However, since the data source is from the Portuguese banking institution, actually we should be more familiar with the social environment of the country so that the encoding part would be done better.

Predictors selection

In the project, we put all providing predictors into the data processing and then training the models. But in the actual business, some predictors probably have no impact on the whole campaign process. If we know more about the business or the details of the campaign, we would recognize some predictors should be removed or concerned with.

Parameter

In the future, we should test and find the best combination of hyperparameters in each model by cross-validation. For example, in SVM_kernel, we are supposed to use cross-validation to find the most proper C (penalty) and gamma (for rbf kernel). Usually, a better combination of hyperparameters can definitely improve the performance of a model. Thus, we can further improve the effectiveness of prediction by updating the hyperparameters.

Time and capacity

The time consumption in some of our models cannot be ignored. For example, the SVM_dual consumes around 7 hours to find a max margin hyperplane for only 6000 records. Even though we accelerate the training process by introducing the simplified SMO algorithm, it still cost too much time in iterations. In the future, we can improve some coding techniques to save the time of modeling.

Reference

Moro, S., Cortez, P., & Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. Decision Support Systems, 62, 22-31.

Moro, S., Laureano, R., & Cortez, P. (2011). Using data mining for bank direct marketing: An application of the crisp-dm methodology.