Modeling and Prediction of deposit based on bank data

Part I: Abstract

This project aims to design and implement several classifiers to predict if a given person will subscribe to a long-term deposit based on the information given in the bank dataset. Basically, the whole project can be divided into three parts: preprocessing, classification and performance evaluation. Six kinds of classifiers, SVM, Naïve Bayes, Logistic regression, K nearest neighbor, perceptron and ANN, are built and tested. Applying the best model, the testing accuracy is 95.30%, the AUC of prediction is 0.9166, and the F1 score can reach up to 0.7662, which means that most of the labels of test sample are predicted correctly. The yes class, represent for individuals who will invest, is successfully predicted at a rate of 77 out of 114, while the people who will not invest is successfully predicted at a rate of 876 out of 886 using 1000 test data.

Part II: Method

Method Overview

This project aims to build several models to predict if a person will subscribe to a long-term deposit. Project can be concluded into four parts. The first step is preprocessing, the information is converted from strings in 'bank_additional.csv' into numeric value, using integer encoding and one-hot encoding. Then all data is normalized and randomly separated into training set (including validation set) and testing set. Secondly, the training set is balanced by applying direct replicated minority class data or SMOTE. The next step implements feature selection according to experience, correlation or sequential forward selection. Thirdly, pattern recognition models are built to classify testing data and the performance is tested by analyzing classification accuracy, ROC curve and F1 score.

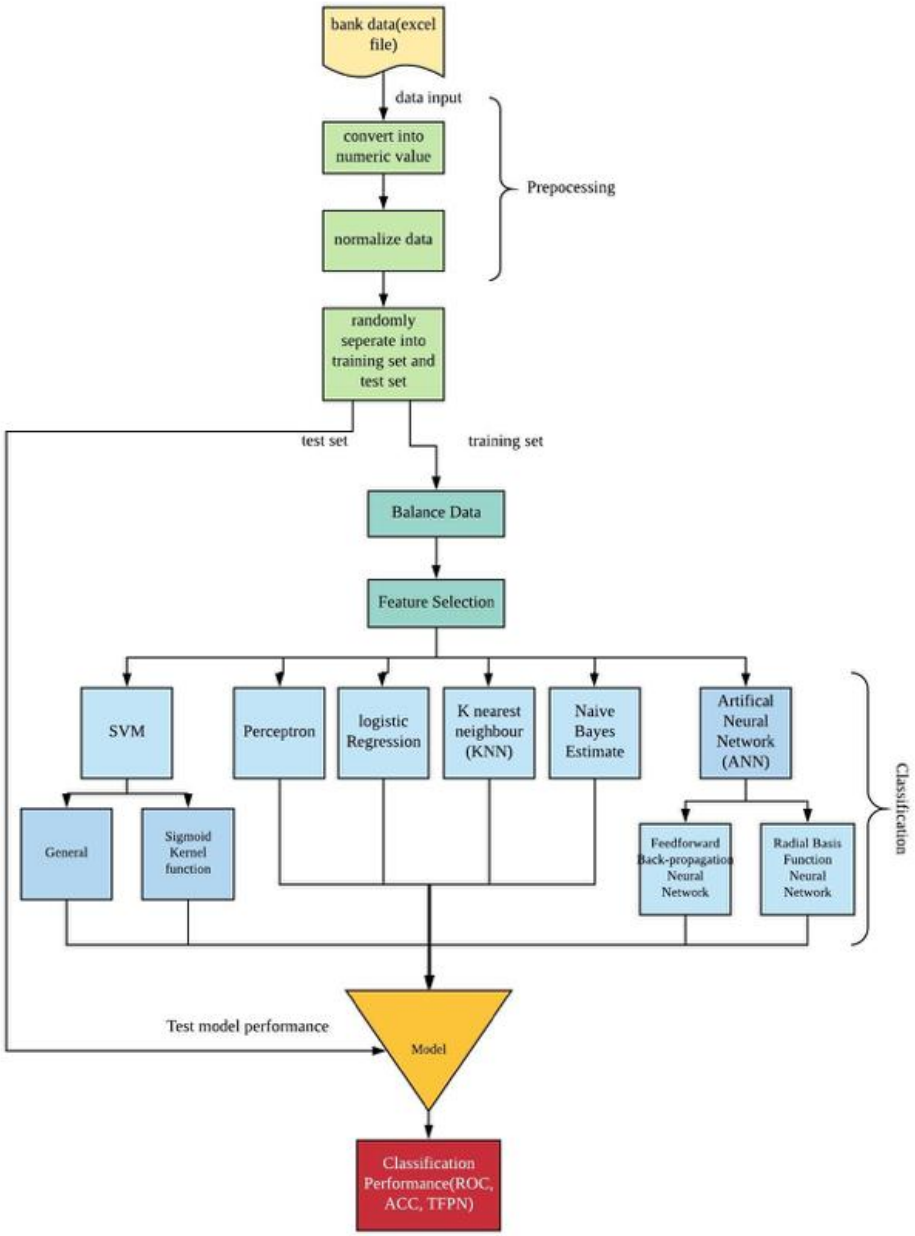

Figure 1: Flowchart of this project.

Preprocessing

1. Data input

Original dataset contains integer number, float number and string. In order to read it from file, MATLAB build-in function `textscan` is used.

Categorical data, represented by string in dataset, need to be encoded. In other words, converted from label value to numeric value, for future preprocessing and classification. If one feature has less than three categories, it is encoded directly using integer encoding by assigning integer value for each category. Housing, loan, contact, poutcome and y are encoded using this technique. Otherwise, according to characteristics of different features, different encoding techniques are applied. For each feature with ordered relationship among categories, integer encoding is enough. For example, marital and education are separately encoded in order according to relationship. However, if

relationship among categories is not clear, integer encoding will bring ordering into consideration when build the classification model and result in poor performance. One-hot encoding can solve this problem by represent each category by a binary variable, where 1 represent for yes and 0 represent for no. In this dataset, job, month and day of week are encoded using one-hot encoding.

Unknown variables, exists in job, education, default, housing and loan. By observing, default contains only one yes. As all the rest are no and default, it can be treated as a useless feature and deleted directly. Unknowns in other features has small quantities, so they are treated as a unique category in job and education features and for housing and loan, unknowns are given a mean of other two categories.

2. Normalization

After data input from dataset, rescaling is applied to each feature by using $X' = \frac{X - X_{min}}{X_{max} - X_{min}}$. By rescaling all value into [0,1], influence of features on feature selection and final classification can be normalized.

3. Randomization

To make sure the model can be applied in generally case, randomizing data before separate into training and test set is necessary. To randomize data, the function `random` is built, which can random assign data into training and test set.

4. Data balance

The bank dataset faces with problem of dataset imbalance, where about 90% data belong to no class and only 10% belong to yes class. Using imbalance dataset may causes poor classification on minority class. Although both undersampling of majority class or oversampling of minority class can both solve dataset imbalance problem, the first approach takes the risk of data loss. Therefore, two oversampling techniques are implemented to solve this problem.

First and easiest way is to replicate the samples from minority class until the number of two class are close.

Except for directly replication data, modern techniques can be applied to deal with dataset imbalance in feature space. SMOTE (Synthetic Minority Over-sampling Technique) is implemented and tested in this project. The algorithm can be concluded as followed:

1. Find k-nearest neighbors belong to same minority class as a data point.
2. Create new data point using the data point X and each of its selected neighbor X_n according to $X_{new} = X + rand(0,1) * |X - X_n|$. So that k new data points can be created.
3. Repeat step 1-2 for each data point in minority class.

SMOTE is achieved using online source code. [1][2]

5. Feature selection

After data input and encoding, original dataset is converted to a 41-dimensional dataset. Using the high-dimensional dataset directly takes the risk of long training time and overfitting. Feature selection is then required to reduce feature dimension by selecting

most important features. Three methods are implemented.
The first method is empirical selection.
The second one is selecting according to correlation. This method aims to select features highly correlated with classes, but less correlated with each other to form a subset of features using in training and testing.
The third method is sequential forward selection. It is based on KNN classification and cross-validation, starting with a blank subset, selecting one best performance feature at each round until given subset size. This is achieved by online source code. [3][4]

Classification

1. SVM with sigmoid kernel function

SVM (Support Vector Machine) is a distribution-free classification method for classifying both 2-class problem and multi-class problem. In SVM, by adding constrains to the criterion function, we find the optimal value which maximize the margin and gives us the decision boundary. In SVM model building, different kernel functions can be chosen for different condition.

$$L_D(\lambda) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j Z_i Z_j k(X_i X_j)$$

Two different kinds of SVM were used, first is Radial basis Kernel function with gamma is 0.5 and C is 0.5. In this part, LIBSVM is used to train and test SVM classifier. 5-fold cross validation is implemented on training set. `cvpartition` is used to implement cross validation. `svmtrain` is used to train the model and `svmpredict` is used in testing.
Second method is Sigmoid kernel function with gamma is 0.5 and C is 5. The model was trained by fitcsvm function in STATISTIC AND MACHINE LEARNING TOOLBOX in Matlab. [5][6]

2. Naïve Bayes

Naïve Bayes classify method based on Bayes theory, can classify 2-class or multi-class problem. The decision rule:

$$P(s_i|x) * P(s_i) > P(s_j|x) * P(s_j) \quad \forall j \neq i \;\; x \in s_i$$

If the prior probability $P(s_i)$ and $P(s_i|x)$ can be found, the model can be built.
By using the `fitcnb` function in STATISTIC AND MACHINE LEARNING TOOLBOX in Matlab, we can directly built the model based on training data and label. And use `predict` to predict the label of test data. [6]

3. Logistic regression

Logistic regression is a regression method for 2-class, binary problem. The output is the probability of each class. The predict function is shown below:

$$\mathrm{p} = \mathrm{p(X)} = \mathrm{S}(X^T\beta) = \frac{1}{1 + \mathrm{e}^{-X^T\beta}}$$

By using the `logisticR` function in STATISTIC AND MACHINE LEARNING TOOLBOX in Matlab, we can directly build the model based on training data and label. And use `predict` to predict the label of test data. [6]

4. K nearest neighbor

K nearest neighbor method is a density estimate method, by counting the nearest k samples of each point, a density map can be derived.

$$P_N(S_i|x) = \frac{P_n(x, s_i)}{\sum_{i=1}^{c} P(x, s_i)} = \frac{k_n^i}{k_n}$$

By using the `KNNI` function, the test set predict label can directly get.

5. Perceptron

Perceptron learning using sequential gradient decent in augment space is applied. Initially, data points are shuffled and reflected. Weight vector is initialized to $\underline{w} = \underline{0.1}$ and learning rate parameter is set to $\eta = 1$. After that, the model is trained for 3000 epochs. Within each epoch, weight vector is updated according to following rules.

$$\underline{w}(i+1) = \begin{cases} \underline{w}(i) + \eta Z_n \underline{X_n}, & if \underline{w}^T Z_n \underline{X_n} \leq 0 \\ \underline{w}(i), & if \underline{w}^T Z_n \underline{X_n} > 0 \end{cases}$$

At the last epoch, perceptron criterion function $J(\underline{w}) = -\underline{w}^T Z_n \underline{X_n} \left[\!\left[\underline{w}^T Z_n \underline{X_n} \leq 0 \right]\!\right]$ is calculated after updating weight vector by each data point. Weight vector generates smallest $J(\underline{w})$ is perceptron model. Finally, testing data is classified using selected weight vector.

6. Artificial neural network

ANN composed of input layer, hidden layer and output layer. Each layer contains several neurons. Each neuron takes weighted combination of pervious layer's output and then pass through activation function to decide its output. A trained model can be formed through these steps. Two ANNs are implemented by MATLAB build-in function for training and testing.

The first one is Feed-forward Back-propagation Neural Network. It used supervised learning by comparing output label with true label and passing results back to update weight vector. In order for a better trained model, several hyperparameters need to be adjusted, for example, learning rate, decay and epochs. Matlab build-in function `newff` is used to form the network architecture, `train` is used to train the network and `sim` is used in testing.

Different from Feed-forward Back-Propagation Neural Network, Radial Basis Function Neural Network has fixed three-layer architecture, where hidden layer aims to map data to a linearly separable space. Less hyperparameters required in this network. Matlab build-in function `newrb` is used to train the network and `sim` is used in testing.

Performance Evaluation Techniques

ROC curve is used to test performance of a classifier by plotting false positive rate verses true positive rate. The larger AUC, the better classification result.

$$true\ positive\ rate = \frac{TP}{TP + FN}$$

$$false\ positive\ rate = \frac{FP}{FP + TN}$$

F1 score which calculated by $2 \times \frac{precision \times recall.}{precision+recall}$is also used to imply classification result. 0.5 is the most optimal F1 score.

Part III: Result and Discussion

1. Feature used

Using sequential forward selection, consconfidx, poutcome, self-employed, previous, December, October, pdays, management, housemaid, April, November, entrepreneur, unknown(job), retired, loan, conspriceidx, May, March are 18 selected features (order matters).

2. Dataset usage

In this project, more than 4400 data samples can be extracted from given file. (after delete unknow samples) In this 4400 samples, we random choose 4000, and separate it by 75:25, using 3000 data samples in training set and 1000 data in test set. Because training set and test set are chosen randomly in each method, it guarantees the model can be general used.

3. Classification result

Table 1: Performance of Radial kernel SVM classifier

Radial based kernel SVM (gamma=0.5, C=5)					Interpretation
Feature select & balance performance	ACC	89.60%	TP	66	In Radial SVM, feature selection slightly reduce the prediction accuracy, but it's still very high. The best performance comes from balance data with all features. Overall, Radial SVM works extremely well on F1 score and ROC curve, it predict a high accuracy on negative data and relatively high on positive data
	AUC	0.7582	TN	830	
	F1 score	0.5593	FP	56	
			FN	48	
All feature & balance performance	ACC	95.30%	TP	77	
	AUC	0.9166	TN	876	
	F1 score	0.7662	FP	10	
			FN	37	
All feature & imbalance performance	ACC	76.10%	TP	48	
	ROC	0.626	TN	713	
	F1 score	0.29	FP	173	
			FN	66	

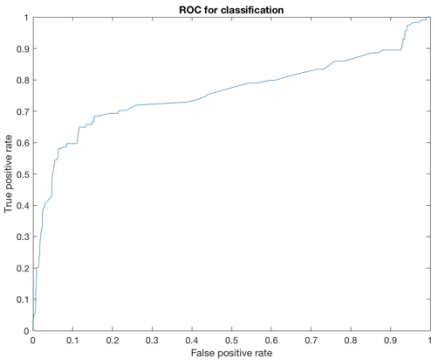

Figure 2: ROC curve for Radial Kernel SVM classifier using feature select and balance data.

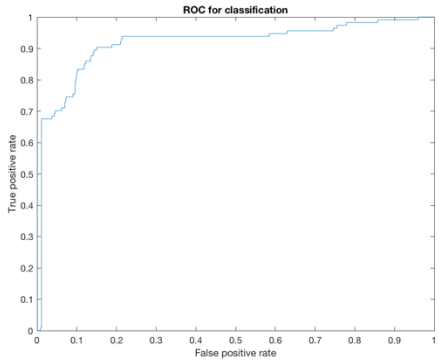

Figure 3: ROC curve for Radial Kernel SVM classifier using all feature and balance data.

Table 2: Performance of Sigmoid kernel SVM classifier

Sigmoid kernel SVM (gamma=0.5, C=-0.5)					Interpretation
Feature select & balance performance	ACC	83.60%	TP	37	In sigmoid SVM, Feature selection improved the prediction performance, but overall this method not fit the data very well, the correct predict positive data (TP) is less than wrong(FN)
	ROC	0.6633	TN	799	
	F1 score	0.329	FP	87	
			FN	77	
All feature & balance performance	ACC	65.80%	TP	68	
	ROC	0.636	TN	590	
	F1 score	0.2863	FP	296	
			FN	46	
All feature & imbalance performance	ACC	76.10%	TP	48	
	ROC	0.626	TN	713	
	F1 score	0.29	FP	173	
			FN	66	

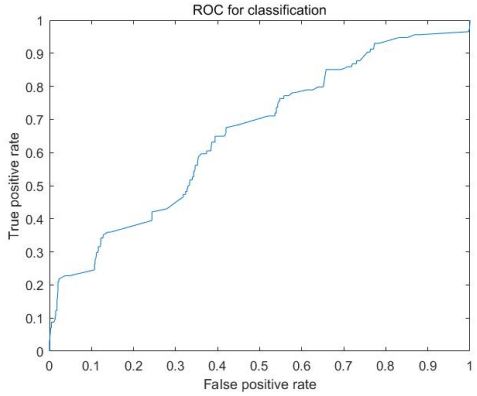

Figure 4: ROC curve for Sigmoid Kernel SVM classifier using feature select and balance data.

Table 3: Performance of Naïve Bayes classifier

Naïve Bayes					Interpretation
Feature select & balance performance	ACC	89.10%	TP	50	In Naïve Bayes method, reduce feature decrease the predict accuracy, and balance is also unnecessary. The raw data actually perform the best classification. Overall, this classifer perform good, it has a high accuracy on negative data(TN), but the positive set is relatively low. (TP)
	ROC	0.748	TN	841	
	F1 score	0.4785	FP	45	
			FN	64	
All feature & balance performance	ACC	88.70%	TP	57	
	ROC	0.794	TN	830	
	F1 score	0.509	FP	56	
			FN	57	
All feature & imbalance performance	ACC	89.70%	TP	55	
	ROC	0.815	TN	842	
	F1 score	0.520	FP	44	
			FN	59	

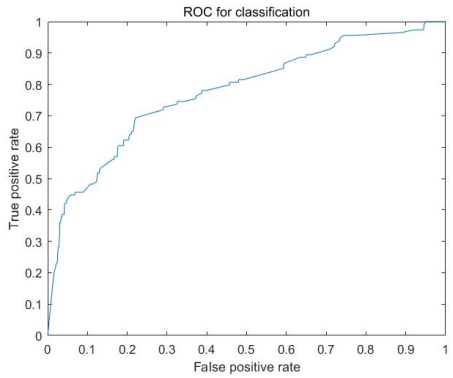

Figure 5: ROC curve for Naïve Bayes classifier using feature select and balance data.

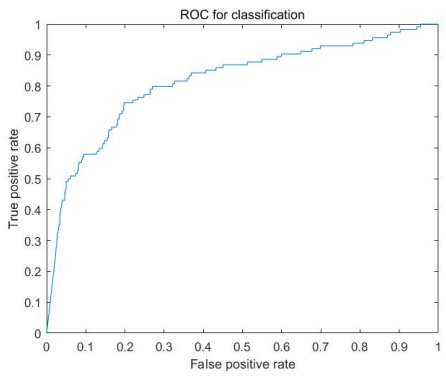

Figure 6: ROC curve for Naïve Bayes classifier using all features and imbalance data.

Table 4: Performance of Logistic Regression classifier

Logistic Regression					Interpretation
Feature select & balance performance	ACC	89.20%	TP	49	In Logistic Regression method reduce feature decrease the predict accuracy, and balance is also unnecessary, Overall, this method is good, it has a high accuracy in negative data and perform ok in positive site, and the ROC is high (0.8).
	ROC	0.7624	TN	834	
	F1 score	0.4757	FP	43	
			FN	65	
All feature & balance performance	ACC	87.50%	TP	67	
	ROC	0.8131	TN	808	
	F1 score	0.5174	FP	78	
			FN	47	
All feature & imbalance performance	ACC	89.10%	TP	58	
	ROC	0.8155	TN	833	
	F1 score	0.5156	FP	53	
			FN	56	

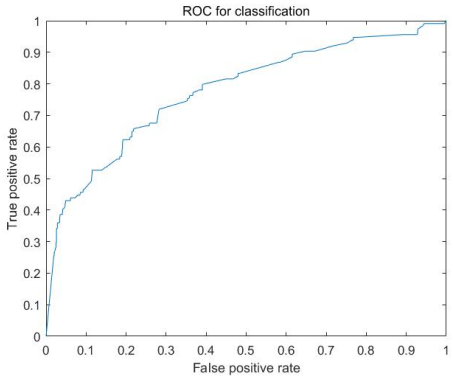

Figure 7: ROC curve for Logistic Regression classifier using feature select and balance data.

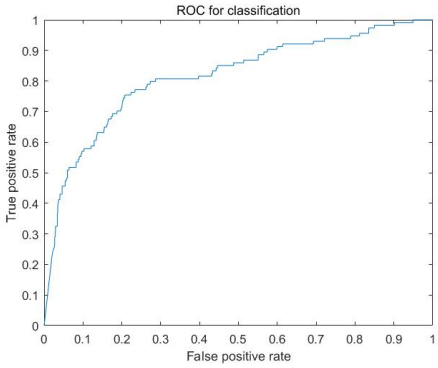

Figure 8: ROC curve for Logistic Regression classifier using all features and imbalance data.

Table 5: Performance of Perceptron classifier

Perceptron					Interpretation
Feature select & balance performance	ACC	78.90%	TP	50	This method is a linear classifier and not suitable for this case.
	AUC	0.6363	TN	739	
	F1 score	0.3215	FP	147	
			FN	64	

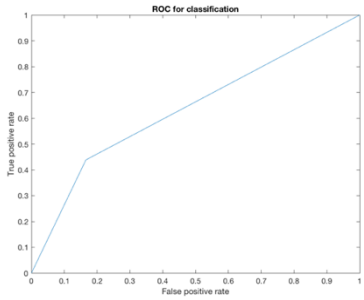

Figure 9: ROC curve for perceptron classifier using feature select and balance data.

Table 6: Performance of Feed-forward Back-Propagation Neural Network

Feed-forward Back-Propagation Neural Network					Interpretation
Feature select & balance performance	ACC	91.10%	TP	64	This method gives an optimal result. Most yes is successfully classified using balanced data.
	AUC	0.8300	TN	847	
	F1 score	0.5899	FP	39	
			FN	50	
All feature & balance performance	ACC	92.20%	TP	72	
	AUC	0.8869	TN	850	
	F1 score	0.6486	FP	36	
			FN	42	
All feature & imbalance performance	ACC	87.20%	TP	52	
	AUC	0.6804	TN	820	
	F1 score	0.4483	FP	66	
			FN	62	

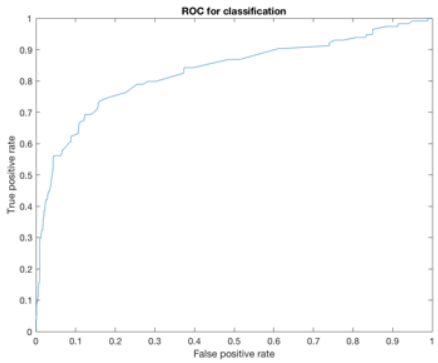

Figure 10: ROC curve for Feed-forward Back-propagation Neural Network classifier using feature select and balance data.

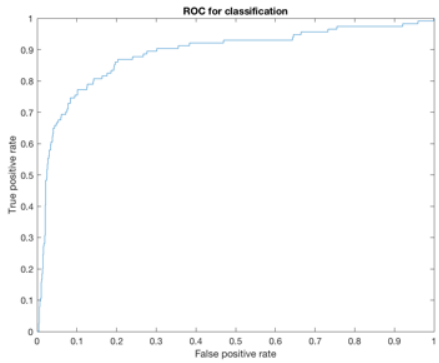

Figure 11: ROC curve for Feed-forward Back-propagation Neural Network classifier using all features and balance data.

Table 7: Performance of Radial Basis Function Neural Network classifier

Radial Basis Function Neural Network					Interpretation
Feature select & balance performance	ACC	89.30%	TP	51	This method gives less optimal result according to accuracy and AUC. However, only half of yes is classified successfully. Balance of dataset doesn't influence final result.
	AUC	0.7849	TN	842	
	F1 score	0.4880	FP	44	
			FN	63	
All feature & balance performance	ACC	89.70%	TP	73	
	AUC	0.8368	TN	824	
	F1 score	0.5864	FP	62	
			FN	41	
All feature & imbalances performance	ACC	89.30%	TP	75	
	AUC	0.8176	TN	818	
	F1 score	0.5837	FP	68	
			FN	39	

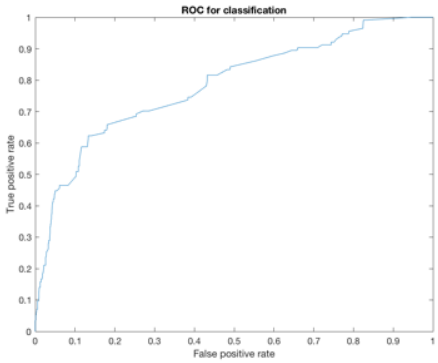

Figure 12: ROC curve for Radial Basis Function Neural Network classifier using feature select and balance data.

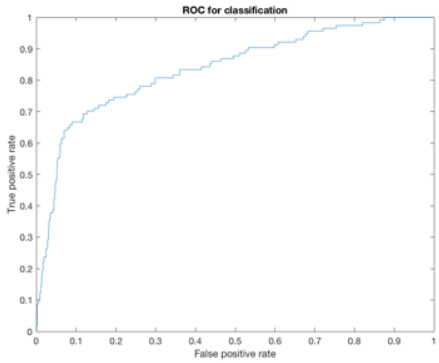

Figure 13: ROC curve for Radial Basis Function Neural Network classifier using all features and balance data.

4. Result Analysis

According to the 7 methods of classification above, the best classification base on ACC and AUC is Radial kernel SVM, which has an accuracy of nearly 0.95 and AUC over 0.9, and it also has the highest accuracy rate on negative data, which is people who will not invest money.

If considering the highest accuracy on positive data, KNN method will be the best, though it has a poor accuracy on negative set, it has the highest accuracy rate on positive set, so it can predict people who will invest money well.

Besides these, Logistic Regression and Artificial neural network are also performed well in both negative and positive data. Meanwhile, perceptron, Sigmoid kernel SVM and Naïve Bayes methods are not performed well (AUC under 0.8).

Part IV: Conclusion and Deficiency

In this project, a given bank dataset is used to predict long tern deposit subscriber after receiving marketing call. MATLAB is used, together with LIBSVM and several online codes. Tian designed randomization, data replicating, empirical feature selection, SVM with sigmoid kernel function classifier, Naïve Bayes classifier, Logistic regression classifier and K nearest neighbor classifier. Jiayue did data input, normalization, data balance using SMOTE, feature selection using correlation and sequential forward selection, perceptron classifier, general SVM classifier and two kinds of ANN classifiers. Performance evaluation part is implemented together by Tian and Jiayue. According to results, the best model is general SVM trained using all features and balanced training dataset. Naïve Bayes, Logistic Regression and Feed-forward Back-propagation Neural Network also gives optimal results.

Part V: Training Method used in news popularity (project 2)

1. Modeling based on best performance method for bank data

Until now, 7 methods have been used in predicting bank data, among them, Logistic regression, SVM and artificial neuron network perform well, so we use these 3 methods to predict the news popularity in the second project.

The preprocessing and follow up procedure are similar to the first project, in the second project, the data are balanced and most features are numerical value, which make the preprocessing easier than the first project. The performance of 3 methods are below:

Table 9: Performance of Feed-forward Back-Propagation Neural Network

Feed-forward Back-Propagation Neural Network					Interpretation
Feature select performance	ACC	67.85%	TP	637	By ANN method, select feature will decrease the accuracy.
	AUC	0.7598	TN	720	
	F1 score	0.6660	FP	453	
			FN	190	
All feature	ACC	73.85%	TP	588	

performance	AUC	0.7802	TN	889	
	F1 score	0.6922	FP	308	
			FN	215	

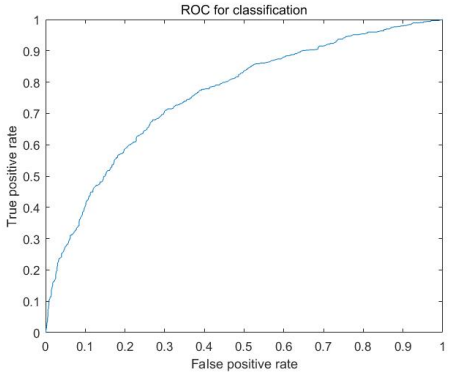

Figure 15: ROC curve for Feed-forward Back-propagation Neural Network classifier using feature select.

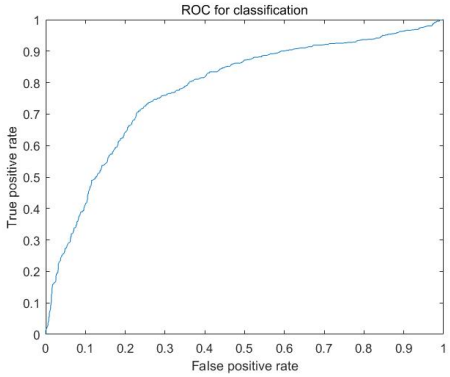

Figure 16: ROC curve for Feed-forward Back-propagation Neural Network classifier using all features.

Table 80: Performance of Radial kernel SVM classifier

Radial based kernel SVM (gamma=1, C=7)					Interpretation
Feature select performance	ACC	66.45%	TP	340	By SVM method, select feature will decrease the accuracy.
	AUC	0.7312	TN	989	
	F1 score	0.5044	FP	173	
			FN	498	
All feature performance	ACC	76.8%	TP	568	
	AUC	0.8457	TN	968	
	F1 score	0.7108	FP	215	
			FN	249	

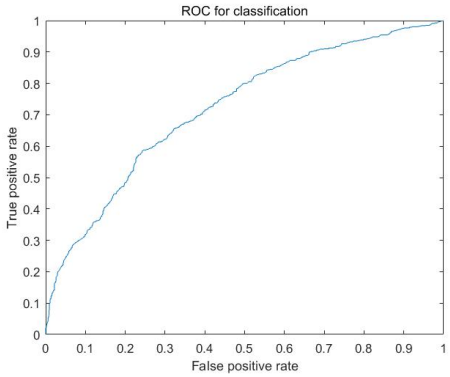

Figure 17: ROC curve for Radial Kernel SVM classifier using feature select

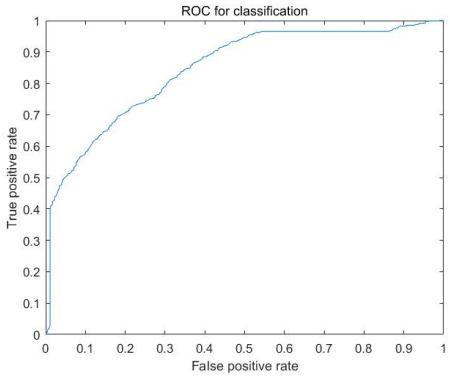

Figure 18: ROC curve for Radial Kernel SVM classifier using all feature

Table 11: Performance of Logistic Regression classifier

Logistic Regression					Interpretation
Feature select performance	ACC	63.25%	TP	585	In Logistic Regression method no big influence on features selection
	ROC	0.7186	TN	680	
	F1 score	0.6492	FP	587	
			FN	148	
All feature performance	ACC	66.5%	TP	610	
	ROC	0.7354	TN	720	
	F1 score	0.6477	FP	478	
			FN	192	

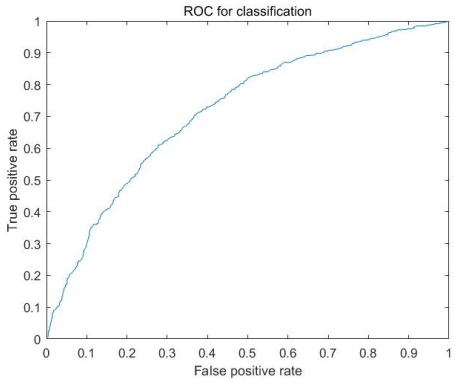

Figure 19: ROC curve for Logistic Regression classifier using feature select.

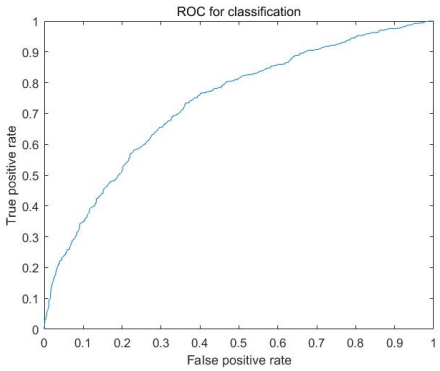

Figure 20: ROC curve for Logistic Regression classifier using all features.

2. Result Analysis

From above results, these 3 methods all have good performance on predict news-popularity problem (accuracy over 65%), higher than the by-chance accuracy (accuracy when predict all sample as the class with more samples), which is nearly 59%, and the best classification method is Radial kernel SVM, which has an accuracy of 76.8%.

References

[1] implement of SMOTE--https://blog.csdn.net/lzy_2016/article/details/56503134
[2] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data" (2013). Available at: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-106#Bib1 (Accessed: 1st May 2018)
[3] Sequential forward selection—https://github.com/skadio/featureSelection/blob/master/SFS.m
[4] D. Ververidis and C. Kotropoulos, "SEQUENTIAL FORWARD FEATURE SELECTION WITH LOW COMPUTATIONAL COST". Available at: https://pdfs.semanticscholar.org/3432/ee16ea67b93d87c939403f4420da65944a2e.pdf (Accessed: 1st May 2018)

[5] libsvm—https://www.csie.ntu.edu.tw/~cjlin/libsvm/ (Accessed: 1st May 2018)
[6] Statistic and machine learning toolbox—
Statistics and Machine Learning Toolbox Documentation