Classification Models for Successful Bank Marketing Prediction

WPI DS502 – Statistical Methods for Data Science

ABSTRACT

This paper will investigate various classification techniques and their accuracy in predicting the outcome of a marketing call during a bank campaign to sign its customers up for a term deposit. The classification models assessed in this paper include Random Forest Classifier (RFC) Support Vector Machine (SVM) and Sequential Neural Network (SNN). Grid Search was implemented to tune hyperparameters in the models and cross validated to ensure that the model was not overfitting to the training set. Model performance using base variables versus transforming the data via Truncated SVD was compared. Models were assessed by their accuracy, precision, and recall.

Both the RFC and SVM models after hyperparameter tuning and cross validation improved the success rate of the marketing campaign from 11.6% to 30% and 25% respectively. Dimensionality reduction using Truncated SVD showed a statically significant improvement in the model performance over using the base variables.

KEYWORDS

Random Forest Classification; Support Vector Machine; Sequential Neural Network; Grid Search CV; Cross-Validation; Truncated SVD; Scikit Learn; TensorFlow

1 INTRODUCTION

When performing a marketing campaign, the goal is to have the highest success rate possible in the timeline set. Without any data analytics, the strategy tends to be contacting every customer that the company has. Depending on the size of the customer base and the number of employees, it may be impossible to contact everyone in the allotted time. When calling every customer with no pre-filtering, the success rate of the campaign can be low and many work hours and money spent making calls that result in failure.

Reducing the number of calls during the campaign by implementing a machine learning algorithm to pre-filter the customers can improve the success rate and reduce the time spent on unsuccessful calls. This pre-filtering is done with a classification algorithm with the result of the call being the target variable. Three different types of classification algorithms were tested in this paper: Random Forest Classifier, Support Vector Machine, and Sequential Neural Network. Each classifier had an initial run using manual tuning to test the validity of the model. Validity was assessed by reviewing the performance of the model based on accuracy, precision, and recall.

After testing the model validity, a grid search can be performed on the hyperparameters to assist in tuning the model to achieve the highest performance. Dimensionality reduction was also implemented and plotted against models using the base variables to see if there is any statistically significant improvement.

After tuning the hyperparameters and validating the models with cross validation and a final validation set, both the RFC and SVM models increased the success rate of the marketing campaign by more than a 2x factor. The number of calls to be made during the campaign was also reduced.

2 PROBLEM STATEMENT & DATA

2.1 Problem Statement

In this paper, the goal is to create a model that predicts whether or not a bank customer will sign up for a term deposit after receiving a marketing call to do so. The motivation behind this task is to reduce the number of calls that a marketing employee needs to make while maintaining a high success rate from their campaigns. By filtering out customers that are predicted to not sign up, the marketing campaign will be more targeted and allow the employees to call more customers with a higher predicted success rate. Reducing and improving the success rate of calls increases the productivity of the marketing teams and allows them to focus on other tasks such as preparing a new ad or a new campaign for another service.

2.2 Data

The data for this paper was obtained from the University of California Irvine dataset repository[1]. The data is collected from a

[1] Available at https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

marketing campaign that the Bank of Portugal performed. The repository contains various sized dataset and external indices. In this paper the full dataset of customers was used.

The data was imported into python and checked for any null or missing values that would need to be imputed. In this case, the bank-full.csv did not include any. The class variable was plotted to get an idea of the distribution of classes.

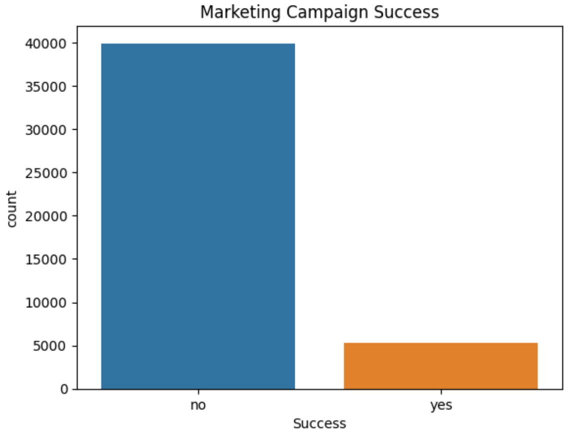

Figure 1: Marketing Success Class Distribution

Overall success rate for the dataset is 11.6%. The number of customers saying no to a term deposit greatly outnumber those that say yes. This will have an impact on the following classification models because the class the model is most interested in is not numerous. Some techniques that can be implemented to improve the model performance include bootstrapping to increase the ratio and adding class weights as a hyperparameter.

2.3 Exploratory Data Analysis

Exploratory Data Analysis (EDA) was performed to get an idea of the distribution of the data. The numerical columns 'age', 'balance', 'campaign', 'day', and 'previous' histograms were plotted. Two of those histograms are shown in Figure 2 and Figure 3. Class labels were also included as a color.

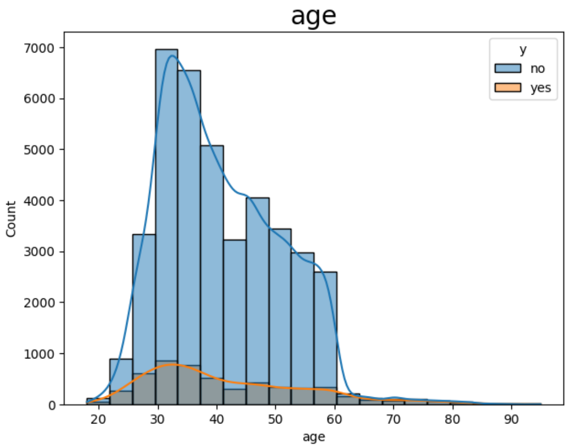

Figure 2: Histogram of Customer Age

The mean age of the customers contacted is 41 and the oldest was 95. Customers over the age of 60 are more likely to say yes to a term deposit than those less than 60. Customers in their early 20s also have a high success rate.

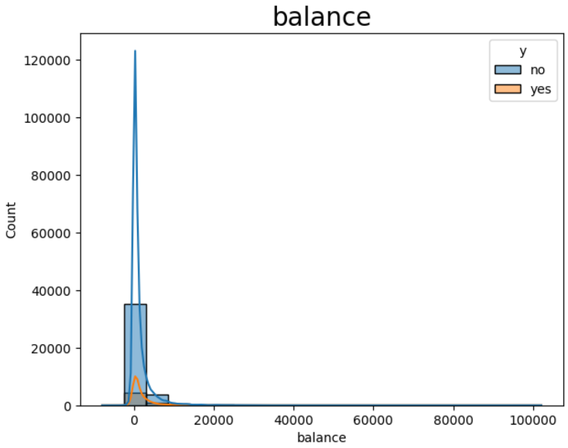

Figure 3: Histogram of Customer Balance

One surprise in this dataset was how little balance many of the customers had in their accounts. The mean balance was 1362 and the lower 75% of customers had a balance of less than 1428. One possible reason for the low success rate in the dataset is that the majority of customers called do not have the ability to put money into a long-term deposit. The other option for this trend in the dataset is that the bank was targeting low balance customers to assist them in increasing their wealth and maintain a healthy customer relationship. Due to the nature of the dataset, these hypotheses cannot be confirmed or denied. The rest of the report will work under the assumption that this is a normal sample set of the bank's customers.

3 CALL SUCCESS PREDICTION

In this paper, three classification models were implemented, Random Forest Classifier (RFC) Support Vector Machine (SVM) and Sequential Neural Network (SNN). A baseline model was created for each of the classifiers to see what their efficacy was before proceeding to more computationally expensive techniques.

3.1 Random Forest Classifier

The base decision tree model runs quickly and provides a base confidence in the decision trees capability in the desired classification. RFC expands on this by creating a set number of decision trees and makes the class selection based on the mean prediction from all the trees. A key aspect of RFC is that the trees created are uncorrelated from each other. This is done by randomly selecting only a subset of the available predictors for each split. Each individual tree follows similar rules of a basic decision tree. The splits at each node are done such that the selected index, such as Gini Index, is reduced.

3.2 Support Vector Machine

A SVM model aims to separate the classes by creating a hyperplane between the class points. When represented in a two-dimensional workspace, a linear hyperplane is a straight line separating X1 from X2. SVM's goal is to find a hyperplane that maximally separates the classes. This means that the distance from the datapoints to the hyperplane will be maximized for both classes. In higher order datasets, a linear SVM may not be ideal. Other kernel functions for the hyperplane include a polynomial, sigmoid, and rbf. RBF is similar to K-Nearest Neighbors and is the initial kernel to use when there is not much known about the dataset [1].

3.3 Sequential Neural Network

SNN is the implementation of a multiplayer neural network using the tensorflow and keras libraries in python. The SNN takes an input vector of variables and builds a non-linear function to predicts the response [2]. The model built for this paper includes multiple hidden layers that reduce the amount of variables down to the single output variable that determines one of the two classes in the classification task. To fit the model, a loss function is used that aims to minimize the loss, or error, between the predicted and actual class labels.

3.4 Grid Search Cross Validation

The RFC and SVM models have various hyperparameters that can be modified to tune the model. Such parameters include number of RFC trees, maximum depth of the tree, SVM kernel type, and SVM regularization parameter. Grid Search Cross Validation (GridCV) is implemented to systematically search through a gid of given parameter values and test all combinations to see if there is improvement in the model. GridCV also incorporated cross validation to get a mean score that can be chosen and outputs the best fit model with the best score.

3.5 Truncated SVD

In the data cleaning and preparation phase of the project, categorical variables were encoded such that they can be passed to the classifiers. Thei resulted in 49 predictors and a sparse matrix of encoded variables. Truncated SVD is a linear dimensionality reduction by truncated singular value decomposition. This technique differs from a Principal Component Analysis as it does not center the data before decomposition, ideal for a sparse matrix [3].

4 PERFORMANCE EVALUATION

In this section, each model will be assessed based on the accuracy, precision, and recall of the model. The goal of increasing the success rate of the campaign is equivalent to the precision of the model [TP/(TP+FP)]. Sacrifices in the precision score will need to be made to ensure that the recall score does not decrease substantially. The importance of recall [TP/(TP+FN)] in this case is to make sure that the marketing team is still calling a large percentage of customers that are predicted to say yes. Therefore, precision and recall will be the focus in defining the final models. The reported values will be the precision and recall for the 'yes' class.

In preparation for fitting the data to the models, a pipeline was created that transformed the categorical columns via sklearn’s One Hot Encoder, scaled and standardized the data via sklearn’s Standard Scaler, and then ran the desired classification model. Two thirds of the dataset was used for fitting the classification models and cross validation and one third was reserved as a final validation set.

4.1 Creating Baseline Models

Initial models for each of the three classifiers were created manually. The goal for the baseline models was to manually adjust inputs such as the hyperparameters and hidden layers inputs. The models were then checked for accuracy, precision, and recall. The summary of these baseline models is in Table 1.

Table 1: Baseline Model Performance

Baseline Model	Accuracy	Precision	Recall
RFC	0.76	0.27	0.61
SVM	0.82	0.34	0.56
SNN	0.83	0.38	0.32

4.2 Grid Search Cross Validation

Both the RFC and SVM models were chosen to run a GridCV on due to the ease of implementation. The parameter grids for both models are captured in Table 2.

Table 2: Grid Search Hyperparameters for RFC (top) and SVM (bottom). Highlighted values are the best fit

Parameter Matrix RFC		
min_samples_split	2	5
n_estimators	100	200
max_depth	25	50
max_features	3	5

Parameter Matrix SVM			
kernel	sigmoid	rbf	
C (Regularization)	0.5	1	2

Stratified Shuffle Split was chosen as the cross-validation technique. This process is similar to a k-fold cross validation but ensures that the ratio between the classes is maintained. Five folds were used to speed up computation time with a 20% validation set. Precision, recall and accuracy were calculated for each fold and parameter combination. Recall was chosen as the refit parameter in GridCV which outputs the model with the best recall score. The values that are highlighted in the parameter tables were the resulting best fit hyperparameters from the Grid Search. The resulting metrics for the two models are in Table 3 and their respective confusion matrices in Figure 4.

Table 3: Model Performance after GridCV using Best Fit Model

Grid Search Model	Accuracy	Precision	Recall
RFC	0.81	0.30	0.46
SVM	0.75	0.25	0.58

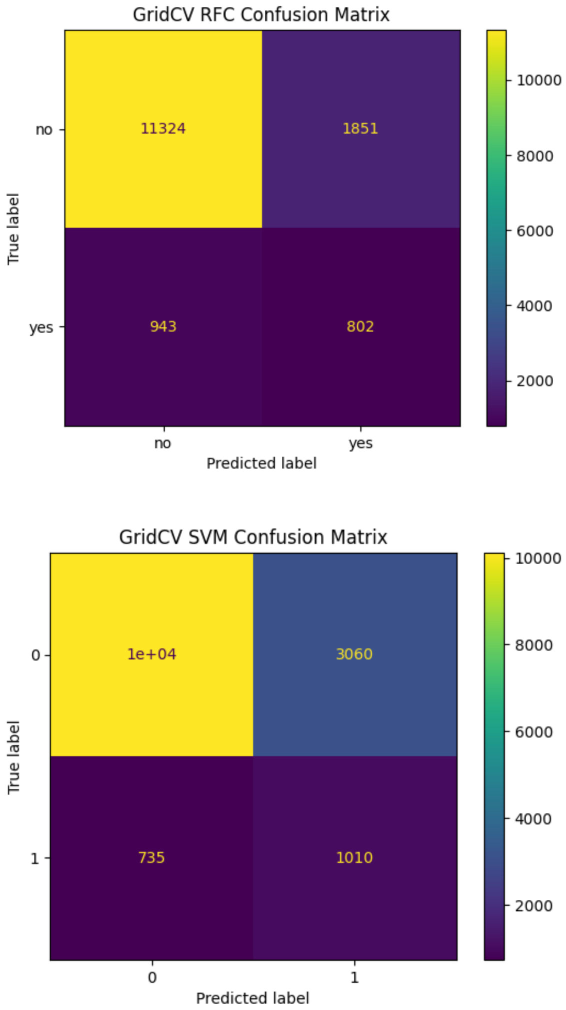

Figure 4: Confusion Matrices for GridCV RFC (top) and GridCV SVM (bottom)

The two models performed similarly when looking at the performance matrix. GridCV SVM did have a larger raw number of successful calls predicted correctly with the tradeoff being an increased number of False Positive calls. The Grid Search algorithm can also be run to return the best model in regards to precision if that is what the marketing team is more concerned with improving.

4.3 Dimensionality Reduction RFC

After encoding the categorical data, there are 50 predictors with many of them being sparce matrices from the encoding process. Truncated SVD was used for dimensionality reduction due to its ability to handle sparce matrices. A model with 25 components was created and their individual and cumulative explained variance plotted.

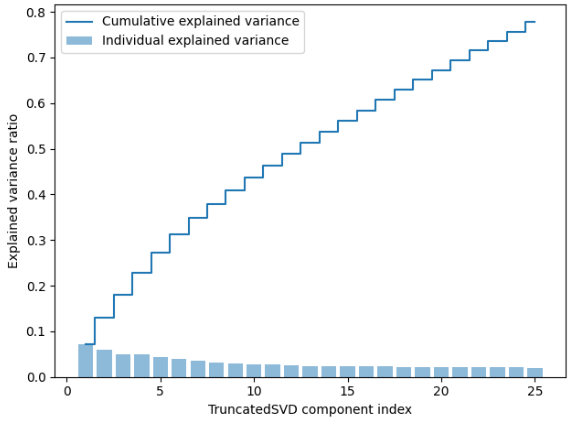

Figure 5: Truncated SVD Individual and Cumulative Explained Variance

The 25 components are close to explaining 80% of the variance in the data. Each individual component's contribution is less than 10% which indicates that the initial variables were not highly correlated.

Using the best fit RFC from GridCV, two models were fitted, one using the initial variables and the other using the reduced dimension components from Truncated SVD. Using a Stratified Shuffle Split with 10 folds and a sample size of 20%, the area under the precision recall curve (AUC) was calculated for each fold. The AUC was plotted for the two models for comparison in Figure 6.

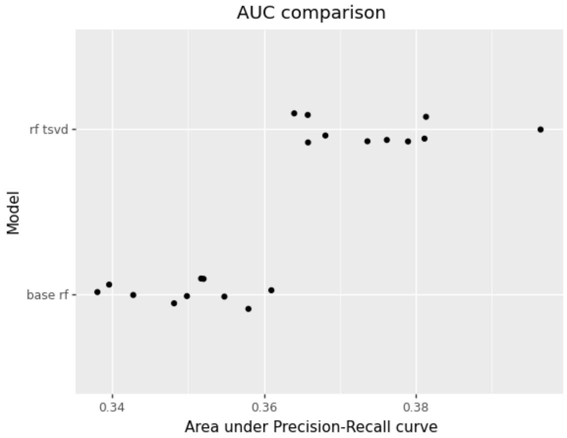

Figure 6: AUC Comparison between using the reduced variables (top) and the base variables (bottom)

The RFC with Truncated SVD did perform better than with the base variables. A two-sided test was tun to check that there is a statically significant difference in the models.

Table 4: Regression Results for Truncated SVD comparison

OLS Regression Results			
Dep. Variable:	auc	**R-squared:**	0.698
Model:	OLS	**Adj. R-squared:**	0.681
Method:	Least Squares	**F-statistic:**	41.55
		Prob (F-statistic):	4.58e-06
		Log-Likelihood:	67.184
No. Observations:	20	**AIC:**	-130.4
Df Residuals:	18	**BIC:**	-128.4
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	0.3496	0.003	124.678	0.000	0.344	0.355
model[T.rf tsvd]	0.0256	0.004	6.446	0.000	0.017	0.034

Omnibus:	1.592	**Durbin-Watson:**	2.199
Prob(Omnibus):	0.451	**Jarque-Bera (JB):**	0.903
Skew:	0.520	**Prob(JB):**	0.637
Kurtosis:	2.948	**Cond. No.**	2.62

The regression result in Table 4 has a p-value less than 0.05 which allows the rejection of the null hypothesis that there was no correlation between the model type and the AUC. There was statistically significant improvement when going from the 50-predictor base model to the 25-predictor Truncated SVD model. Results for the models were then plotted on a precision-recall curve in Figure 7.

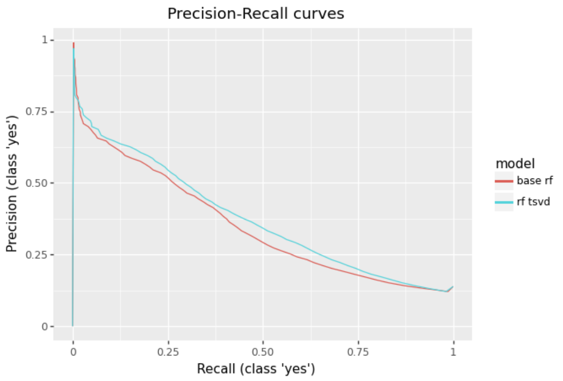

Figure 7: Precision-Recall Curves for Dimensionality Reduction Comparison

Neither model performed extremely well. The ideal precision-recall curve would not see a sharp decrease in precision with little increase in recall. This is due to the large class imbalance in the model and the difficulty in predicting the minority 'yes' class.

5 CONCLUSION

Three initial models were created for comparing performance at predicting if a customer at a bank will say yes to signing up for a term deposit. Random Forest Classifier (RFC), Support Vector Machine (SVM), and Sequential Neural Networks (SNN) were tested. The three models had similar accuracy. SNN had the highest precision but the tradeoff was lower recall compared to RFC and SVM.

Grid Search Cross Validation (GridCV) was performed on the RFC and SVM models to test a matrix of hyperparameters to assist in tuning the models. Each combination of parameters was tested and verified with Stratified Shuffle Split cross validation with 10 folds.

GridCV resulted in an optimal RFC and SVM model based on the highest obtained average recall. RFC had higher recall with a smaller forest and trees with a lower limit on size. The optimal SVM model used the sigmoid kernel and a larger regularization parameter.

Comparing the two models, GridCV RFC resulted in an accuracy of 81%, a precision of 30%, and a recall of 46%. GridCV SVM resulted in an accuracy of 75%, a precision of 25%, and a recall of 58%. Both models have a higher success rate (precision) than the base dataset of 11.6%. Although some actual 'yes' customers were missed, the overall reduction in calls needed results in a more targeted campaign.

The variables were transformed using Truncated SVD to reduce dimensionality and test if there is any change in performance of the model. Using the best fit from GridCV RFC, the best fit performance was compared to the same model using the transformed dataset. Using a two-sided regression, it was confirmed that using Truncated SVD resulted in a statistically significant difference in the model performance.

Future work can include running a larger grid search for tuning hyperparameters. Incorporating the reduced dimension dataset into the grid search is also predicted to increase performance of the model. Future models can also adjust the decision boundary of the probability for class selection to test for increased performance.

REFERENCES

[1] Sreenivasa, S. (2020) *Radial basis function (RBF) kernel: The go-to kernel*, *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a

[2] Tibshirani, R. *et al.* (2021) *An introduction to statistical learning: With applications in R*. Springer. ISBN: 1071614177

[3] *Sklearn.decomposition.truncatedsvd scikit*. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html