Using Machine Learning to Improve the Results of Direct Bank Marketing

Oh Tien Cheng

Abstract

Direct Marketing is a form of marketing that relies on direct communication to individual customers to sell a product or a service. For direct marketing to be effective, it is crucial that only customers that are likely to be persuaded by such forms of marketing be contacted, so as to conserve campaign costs and prevent potential customers from being irritated. In this paper, we propose a supervised learning approach to improve the efficiency of a direct marketing campaign through the use of a classifier to predict the response of a potential customer. We attempt to make use of algorithms for over-sampling and select from several machine learning models to build a highly effective classifier which can accurately predict if a target of a direct marketing attempt will purchase a term deposit loan.

1 Introduction

In this paper, we attempt to make use of a supervised learning approach to improve the direct marketing process. We aim to create a machine learning classifier capable of predicting if a target of a direct marketing call by a bank would be willing to purchase a term deposit.

1.1 Direct Marketing

Direct marketing is a form of marketing that relies on direct communication to individual customers rather than through a third party advertising media agency.Kenton and Anderson (2020). This form of marketing was identified and defined by Lester Wunderman in 1967 (McAteer (2019)). This process usually involves a company directly using social media, email, mail, direct calling and messaging to deliver their messaging and sales pitches. For direct marketing to be effective, the campaign needs to focused on clients who are most likely to respond favourably to the sale pitch. Otherwise, the campaign risks annoying its audience, resulting in a less favourable view of the company.

1.2 Machine Learning

Machine Learning is defined by Mitchell (1997) as the study of computer algorithms that improve automatically through experience. Applications range from data mining programs that discover general rules in large data sets, to information filtering systems that automatically learn users' interests.

1.3 Applications of Machine Learning in the Banking Industry

In recent years, investments by financial institutions like banks in the use of machine learning has increased. Traditionally, the use of machine learning algorithms in financial institutions has been focused on use cases like fraud monitoring and customer interaction via chat bots. (Digalaki (2021)) One area that has not received as much attention, however, is the area of direct marketing. In the banking industry, direct marketing is used to promote new services and products. One such product is a term deposit, which is the product of interest in our data set. A term deposit is a type of fixed investment where a customer puts his or her money in a bank for a fixed period, and in return, receives a higher interest rate as compared to a savings account. (Scott (2021)). While not as much attention has been paid to the applications of machine learning for direct marketing, direct marketing can still be an important tool for banks to secure new customers. In a study by tilo et al. (2020) on banks in Nigeria, they found evidence to suggest a significant between direct marketing and the patronage of customers.

2 Related Works

In the past, there have been several related works regarding the use of machine learning for predicting the success of direct marketing.

An early investigation of this problem was by Ling et al. (1998), who used three data sets, one from a Canadian bank, another from a life insurance company and the last from a company that ran a bonus program. They compared a Naive Bayes algorithm and a C4.5 Decision Tree algorithm in conjunction with a random oversampling technique, and found that performing oversampling greatly improved the performance of a Decision Tree, but had no significant impact on the Naive Bayes algorithm. In particular, they discovered that re-sampling the training data to have a 1-1 ratio of positive and negative classes most improves the performance of the final model.

Another investigation of a data mining approach for this problem was by Moro et al. (2011) who collected real-world data from a marketing campaign of a Portuguese bank. They evaluated three algorithms: Naive Bayes (NB), Decision Trees (DT) and Support Vector Machine (SVM), and performed manual feature selection on the data collected to reduce the number of features from 59 to 29. The best model was found to be the Support Vector Machine, which obtained an AUC score of 0.938.

Subsequently, Moro et al. (2014) collected a new set of data from a Portuguese retail bank, related to telemarketing calls. They utilized a semi-automatic feature selection technique, based on business knowledge and forward selection, to reduce the set of features from 150 features to 22 features. In addition, they compared four models: Logistic Regression (LR), Decision Trees (DT), Neural Network (NN), and Support Vector Machine (SVM). They found that a NN performed the best, with an AUC score of 0.8. The data set used in the paper was then subsequently contributed to the University of California, Irvine Machine Learning Repository.

In our paper, we attempt to expand on these previous works, through applying new learning algorithms and over-sampling techniques to improve on their past results.

3 Methodology

3.1 Data Set

The data set, first collected by Moro et al. (2011), has been made publicly available on the University of California, Irvine Machine Learning Repository. The data is from a direct phone marketing campaign of a Portuguese retail bank. It contains the attributes of customers which the bank has contacted, and information on the success of the call in getting the client to subscribe a term deposit with the bank. The data set consists of 17 attributes, and 41188 examples, ordered by date. The features in the data have already been reduced through feature selection by the contributors of the paper. (Moro et al. (2011)) The data set attributes are shown in Table 1.

3.2 Exploratory Data Analysis

Before we create our machine learning models, we will first perform an exploratory data analysis on our data. Exploratory data analysis is an approach to data analysis that employs a number of different techniques to understand the underlying structure of data, discover important variables and their relationships, detect outliers and suggest suitable models for further analysis. (Hinterberger (2009))

3.2.1 Target Variable

We note that the distribution of the target variable is highly imbalanced (see Figure 1). The negative class comprises 88.3% of the examples provided. This means that most clients contacted did not subscribe to a term deposit.

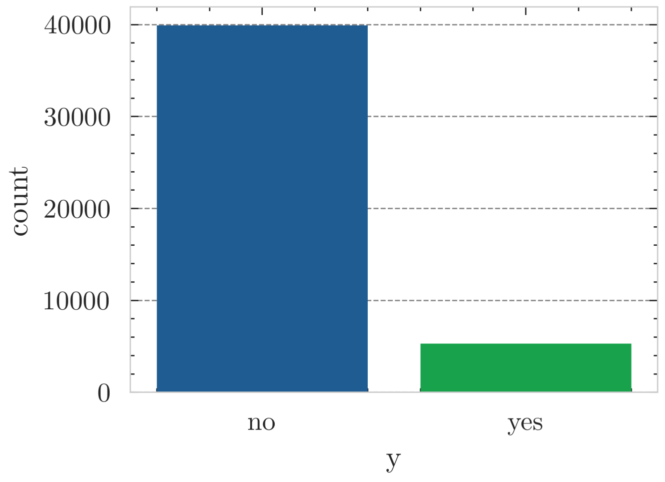


Figure 1: Target Variable Distribution

3.2.2 Quantitative Variables

From analyzing the histograms of the quantitative variables (see Figure 2) and descriptive statistics (see Table 2, we note that:

- With the exception of day, most quantitative features had mean values larger than the median, suggesting that their distributions are positively skewed.
- The frequency of calls per day is cyclical, with there being three main periods of the month where the frequency of calls increases
- The middle 50% of all customers are of ages 33 to 48. The oldest customer is 95. This suggests most customers are middle aged, working adults.
- Some customers had a negative balance in their account, but the middle 50% had an average yearly balance of 72 to 1428 euros
- At least 75% of all clients had never been contacted by the bank for any campaigns. At least one client had not been contacted for more than two years.

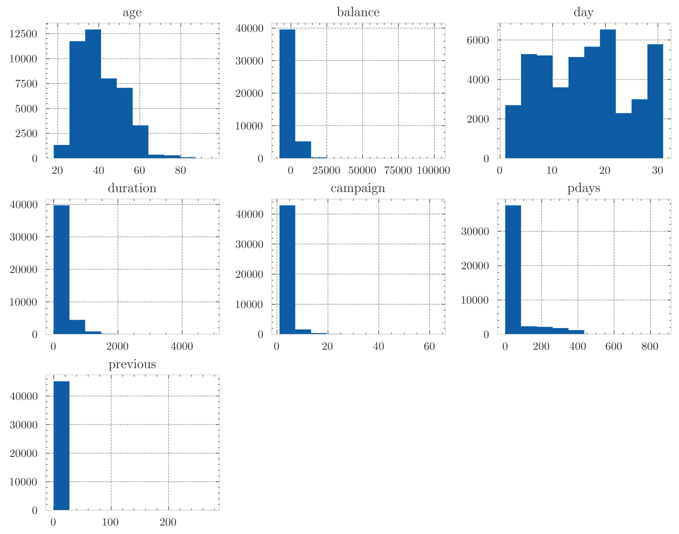


Figure 2: Histograms of Quantitative Variables

3.2.3 Qualitative Variables

From analyzing the descriptive statistics (see Table 3) of the qualitative variables, we note that:

- A large proportion of customers (51.3%) only have a Secondary School education.
- Almost all (98.2%) customers have never defaulted on their credit before. Due to the low variance of this feature, it may not be a useful predictor of the target label.
- The most common form of contact is by cellular (mobile phone) instead of by a wired telephone.
- Most have never been contacted by the bank before, as for most, the outcome of any previous direct marketing campaign is unknown.

We also analyze the count plots of the various categorical features. From figure 3 we note that most calls in the data take place in the middle of the year, between May and August.

Attribute	Data Type	Description
age	Quantitative (Discrete)	Age of the Customer
job	Qualitative (Nominal)	Type of Job
marital	Qualitative (Nominal)	Martial Status
education	Qualitative (Nominal)	Education Level of Customer
default	Qualitative (Nominal)	Does Customer Have Credit In Default?
balance	Quantitative (Continuous)	Average Yearly Balance of Customer
housing	Qualitative (Nominal)	Does Customer Have a Housing Loan?
loan	Qualitative (Nominal)	Does Customer Have a personal Loan?
contact	Qualitative (Nominal)	How was contact made with the customer?
month	Qualitative (Nominal)	What was the month in which the customer was last contacted?
day	Qualitative (Nominal)	What was the day in which the customer was last contacted?
duration	Quantitative (Continuous)	When was the client last contacted?
campaign	Quantitative (Discrete)	Number of contacts performed before this campaign and for this client
pdays	Quantitative (Discrete)	Number of days passed after the client was last contacted from a previous campaign
previous	Quantitative (Discrete)	Number of contacts performed before this campaign and for this client
poutcome	Qualitative (Nominal)	Outcome of the previous campaign
y	Qualitative (Nominal)	Target Variable; Has the client subscribed for a term deposit?

Table 1: Dataset Attributes

	age	balance	day	duration	campaign	pdays	previous
mean	40.94	1362.27	15.81	258.16	2.76	40.2	0.58
std	10.62	3044.77	8.32	257.53	3.1	100.13	2.3
min	18	-8019	1	0	1	-1	0
25%	33	72	8	103	1	-1	0
50%	39	448	16	180	2	-1	0
75%	48	1428	21	319	3	-1	0
max	95	102127	31	4918	63	871	275

Table 2: Descriptive Statistics of Quantitative Variables (to 2 d.p.)

	job	marital	education	default	housing	loan	contact	month	poutcome
unique	12	3	4	2	2	2	3	12	4
top	blue-collar	married	secondary	no	yes	no	cellular	may	unknown
freq	9732	27214	23202	44396	25130	37967	29285	13766	36959

Table 3: Descriptive Statistics of Qualitative Variables (excluding target variable)

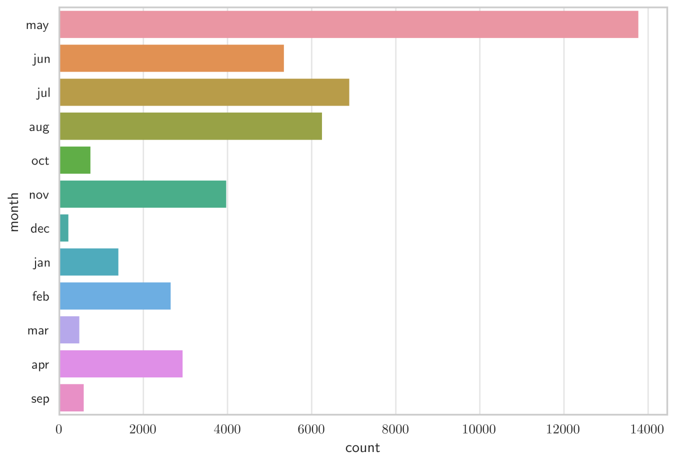


Figure 3: Count Plot of Months

3.2.4 *Bi-variate Analysis*

3.2.5 *ϕ_k Correlation*

To measure the relationship between the variables in the data set, we will use the ϕ_k coefficient. ϕ_k is a new correlation coefficient introduced by Baak et al. (2019), which expands on Pearson's χ^2 statistic. It works consistently with categorical, ordinal and continuous variables, and is bounded to the range [0, 1]. Hence, this makes it useful for our data set, which has a mix of numerical and categorical variables. To create a ϕ_k correlation matrix, we make use of the *phik* library in Python, and create a heat-map from the correlation matrix using the *Seaborn* library. The output is shown in figure 4. From the results of the correlation heat map, we note that:

- *poutcome* has the strongest correlation (0.46) with the target label. This means that the previous outcome of the previous campaign of the customer has the greatest association with the success of the direct marketing call.
- *duration* has the second highest correlation (0.36) with the target label. However, since duration is only recorded after the call has ended, when the outcome of the call is already known, it cannot be used as an actual predictor in our problem.
- *marital, default, balance, contact, campaign, previous* all have very weak correlation (<0.1) with the target label.
- There is a strong correlation between the *job* feature and the *education* feature.

3.3 *Models*

For this paper, we will be testing various models, implemented in the Python *scikit-learn* library (Pedregosa et al. (2011)) on the data set. The models to be evaluated are:

- Logistic Regression (LR)

- Support Vector Machine with Linear Kernel (SVM)
- Decision Trees (DT)
- Random Forests (RF)
- Gradient Boosting Trees (GBM)
- Histogram-based Gradient Boosting Trees (HistGBM)

In addition, we will also compare our models against a baseline mode value predictor.

3.3.1 Baseline Mode Value Predictor

As a baseline of comparison, we will use the *DummyClassifier* in *scikit-learn*, which predicts the most frequent value in the data set.

marital
married
single
divorced
single

Table 4: Marital Feature (sample of 4 rows)

marital_married	marital_single	marital_divorced
1	0	0
0	1	0
0	0	1
0	1	0

Table 5: Marital Feature (after one hot encoding)

3.4 Data Pre-processing

The overall data pre-processing workflow is shown in Figure 5. A more detailed description of the steps involved is given in subsequent sub-sections.

This workflow is implemented via the *Pipeline* class in the *imbalanced-learn* library in Python, which extends *scikit-learn* with functionality to perform re-sampling with cross validation, without mutating the testing fold.

3.4.1 Feature Scaling

Feature scaling is a form of feature engineering that involves individually changing the scale of quantitative features, such that all features are on a similar scale. This can improve the performance of models such as Logistic Regression which are sensitive to the scale of the input features. (Zheng and Casari (2018)). Due to the presence of many outliers in the data, we will utilize Robust Scaling to scale our data. Robust Scaling scales our data by subtracting the median of each feature from our data, and dividing by the inter quartile range of each feature.

$$X_{scaled} = \frac{X - median(X)}{IQR(X)}$$

In *scikit-learn*, robust scaling is implemented via the *RobustScaler* transformer. (Pedregosa et al. (2011))

3.4.2 Encoding Categorical Features

The data contains qualitative variables that are encoded as strings. For this data to be utilized by our models, it is necessary to encode them as numeric data. In this experiment, we attempt two methods of encoding categorical features: One Hot Encoding and Native Handling of Categorical Variables Through Histogram Based Gradient Boosting.

One Hot Encoding is a method for encoding nominal variables by converting each level of a categorical variable into a separate Boolean feature denoting the presence or absence of that level. For instance, in the data, the martial feature is represented as follows in Table 4. After one hot encoding the feature, it is now represented as follows in Table 5.

While one-hot encoding preserves the nominal nature of the data by not placing any implicit ordering of the numerical data, it also greatly increases the dimension of the data set in the case where there are many levels for each categorical variable, and many categorical variables, causing the data to become more sparse. This problem is known as the curse of dimensionality (Hastie et al. (2009b)), where the complexity of a learning algorithm can grow exponentially with the dimension, such that the size of the training set has to grow exponentially as well to prevent the model from over fitting the data. In a typical tree based algorithm like Random Forests, a tree built on one-hot encoded features tends to be unbalanced and needs to grow very deep to achieve good accuracy, thus increasing the complexity of the model. (Ke et al. (2017))

Another method for encoding categorical data is to utilize the native support for categorical features in the Histogram-based Gradient Boosting Tree . The *scikit-learn* implementation, which follows the implementation by LightGBM (Ke et al. (2017)), first requires the categorical features to be encoded as integers. Each level in the categorical features are then binned to form a histogram, sorted according to a heuristic based on the predictions of a base estimator . The model is then able to form a split on the categorical variable based on the histogram. This preserves the nominal nature of the variable, since the levels are not ordered arbitrarily, while reducing the number of dimensions needed to represent the data.

3.4.3 Oversampling

To deal with the imbalanced classes in the data set, will augment our training data to provide our learning algorithms more examples from the minority class. Two oversampling techniques are used in this paper: Random Over-sampling and SMOTE-NC. These techniques are implemented using the *imbalanced-learn* library in Python. (Lemaître et al. (2017))

Random Over-sampling is a simple technique for generating new samples of the minority class. It works by randomly sampling with replacement existing samples in the training data. (Ling et al. (1998)) This technique is implemented in the *imbalanced-learn* library as the *RandomOverSampler* transformer.

SMOTE stands for Synthetic Minority Over-sampling Technique, and was introduced by Chawla et al. (2002). In their paper, they showed that random oversampling of the minority class can cause the model to over-fit to the data. To solve this, they proposed an over-sampling technique, SMOTE, that creates synthetic samples using a k nearest neighbours algorithm. These synthetic examples cause the learning algorithm to create a larger and less specific decision boundary when compared to random over-sampling, thus reducing the risk of over-fitting. However, one drawback of the method is that it is only designed to handle continuous features.

To solve this, Chawla et al. (2002) introduced Synthetic Minority Over-sampling Technique-Nominal Continuous (SMOTE-NC), which can deal with a mix of continuous and nominal features, by using the most frequent category of the nearest neighbours.

3.5 *Evaluation Methodology*

To evaluate our learning algorithms, we split our data set into a training and test set. (Raschka (2020)) The our models are selected based on the cross-validated scores, and the final algorithms are evaluated on the independent test set, which acts as a unbiased estimator of our model performance on unseen data.

The evaluation metric used to evaluate our models will be balanced accuracy.

3.5.1 *Confusion Matrix*

A confusion matrix summarizes the performance of a model on test data as a contingency table. Ting (2017) The rows represent the actual class of a test example while the columns represent the predicted class of a test example by the classifier. For a binary classification problem, the confusion matrix (see Table 6) can be divided into four quadrants. From the confusion matrix, it is possible to obtain other metrics of a classifier's performance, such as accuracy, precision, and recall.

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

Table 6: Confusion Matrix

3.5.2 *Accuracy*

The accuracy of a classifier is the proportion of correctly classified samples out of the entire sample. It is given by

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

While accuracy is an easily understandable metric, it is not always the best option. In the case of this data set, where there is a heavy class imbalance, accuracy as a metric is not useful in telling us the performance of a model. For instance, a baseline model that always predicts the negative class would obtain 88.3% accuracy, despite being useless as a model for predicting the success of a direct bank marketing attempt.

3.5.3 *Balanced Accuracy*

To solve this issue, we adopt a different metric known as balanced accuracy. Balanced Accuracy is given by

$$Balanced\ Accuracy = \frac{1}{2}(\frac{TP}{TP + FN} + \frac{TN}{TN + FP})$$

In effect, it is equivalent to the mean of the True Positive Rate and True Negative Rate of the classifier. When using balanced accuracy, a model would need to good at predicting both the positive and negative class to obtain a high score. Hence, it is more useful in this scenario where we are interested the model being able to predict both classes well.

3.5.4 *Cross Validation*

To select the best learning algorithm, we make use of Stratified K-fold cross validation to obtain a robust estimate of our model prediction performance. Stratified K-fold cross validation (see Algorithm 1) is an extension of the K-fold cross-validation algorithm (Hastie et al. (2009a)) .

Algorithm 1: Stratified K-Fold Cross Validation

```
Data: Let $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$ be the data set with $m$ training
      examples
Input: $K$ = Number of Folds
Divide $S$ into $k$ cross-validation folds (groups), such that
  each group contains approximately the same percentage of
  samples of each class as the data set $S$;
for $k = 1, \dots, K$ do
    Set fold $k$ as the test fold;
    Train the model on the remaining non-test folds;
    Evaluate the model on the test fold;
end
Calculate the average performance over $K$ folds;
Output: Average Cross Validation Score
```

For this paper, we have set the value of K to be 5.

4 Discussion

4.1 *Results*

4.1.1 *Initial Comparison*

As a point of comparison, we record the performance of our models when no re-sampling techniques are applied in Table 7. We note that out of all the models, a decision tree has the best performance in terms of balanced accuracy on the cross validation folds.

Model	Training		CV	
	Acc	Bal Acc	Acc	Bal Acc
Baseline	0.88302	0.5	0.88302	0.5
LR	0.89241	0.89209	0.84674	0.58090
Linear SVM	0.56865	0.84865	0.84674	0.56569
DT	**1**	**1**	0.82836	**0.61565**
RF	0.99995	0.99979	0.89184	0.60300
GBM	0.89861	0.60622	**0.89424**	0.59524
HistGBM (OHE)	0.90566	0.63633	0.89350	0.60528
HistGBM (Native Encoding)	0.90794	0.64318	0.89399	0.60709

Table 7: Model Results (before Oversampling)

4.1.2 *After Over-Sampling*

The results after various oversampling techniques are utilized are detailed in Table 8. From the results, we note that the approach

Model	Training		CV	
	Acc	Bal Acc	Acc	Bal Acc
LR (Random Oversampling)	0.69941	0.74036	0.73855	0.69490
DT (Random Oversampling)	**1**	**1**	0.83532	0.60668
RF (Random Oversampling)	**1**	**1**	**0.88526**	0.63639
HistGBM (OHE & Random Oversampling)	0.79600	0.84974	0.82714	0.73347
HistGBM (Native Encoding & Random Oversampling)	0.80609	0.85561	0.83015	**0.73395**
LR (SMOTE-NC)	0.66619	0.72433	0.66301	0.72224
DT (SMOTE-NC)	1.00000	1.00000	0.80806	0.62261
RF (SMOTE-NC)	0.99979	0.99994	0.86397	0.67529
HistGBM (OHE & SMOTE-NC)	0.67814	0.89581	0.88509	0.65547
HistGBM (Native Encoding & SMOTE-NC)	0.69109	0.89582	0.88271	0.66243

Table 8: Model Results (after Oversampling)

which results in the best balanced accuracy score is through using the Histogram Gradient Boosted Tree, with a balanced accuracy score of 0.7395.

4.1.3 Final Model Evaluation

Finally, we use the best approach of HistGBM with Native Encoding and Random Oversampling, training a model on our entire test set and evaluating on an independent test set. The confusion matrix obtained from the final evaluation is shown in Figure 9.

		Predicted	
		Positive	Negative
Actual	Positive	6851	1134
	Negative	404	654

Table 9: Confusion Matrix

The final balanced accuracy score obtained is 0.738, which is very close to the cross validation score obtained, suggesting the final model is able to generalize well.

5 Conclusions

In conclusion, we have investigated an approach for improving the efficiency of a direct marketing campaign through machine learning. Our final approach is based on random oversampling of the training set, and a Histogram Based Gradient Boosting Tree. We have found that this approach gives us the best balanced accuracy score, and generalizes well to unseen data.

References

M. Baak, R. Koopman, H. Snoek, and S. Klous. A new correlation coefficient between categorical, ordinal and interval variables with pearson characteristics, 2019.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, Jun 2002. ISSN 1076-9757. doi: 10.1613/jair.953. URL http://dx.doi.org/10.1613/jair.953.

E. Digalaki. The impact of artificial intelligence in the banking sector & how AI is being used in 2021, Jan 2021. URL https://www.businessinsider.com/ai-in-banking-report.

T. Hastie, R. Tibshirani, and J. Friedman. Model Assessment and Selection. In T. Hastie, R. Tibshirani, and J. Friedman, editors, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, pages 219–259. Springer, New York, NY, 2009a. ISBN 978-0-387-84858-7. doi: 10.1007/978-0-387-84858-7_7. URL https://doi.org/10.1007/978-0-387-84858-7_7.

T. Hastie, R. Tibshirani, and J. Friedman. Overview of Supervised Learning. In T. Hastie, R. Tibshirani, and J. Friedman, editors, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, pages 9–41. Springer, New York, NY, 2009b. ISBN 978-0-387-84858-7. doi: 10.1007/978-0-387-84858-7_2. URL https://doi.org/10.1007/978-0-387-84858-7_2.

H. Hinterberger. *Exploratory Data Analysis*, pages 1080–1080. Springer US, Boston, MA, 2009. ISBN 978-0-387-39940-9. doi: 10.1007/978-0-387-39940-9_1384. URL https://doi.org/10.1007/978-0-387-39940-9_1384.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

W. Kenton and S. Anderson. Direct Marketing: What You Need to Know, Oct. 2020. URL https://www.investopedia.com/terms/d/direct-marketing.asp.

G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL http://jmlr.org/papers/v18/16-365.html.

C. Ling, , C. X. Ling, and C. Li. Data mining for direct marketing: Problems and solutions. In *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98*, pages 73–79. AAAI Press, 1998.

O. McAteer. Lester Wunderman passes away aged 98, Jan. 2019. URL https://www.campaignlive.co.uk/article/lester-wunderman-passes-away-aged-98/1522728?utm_source=website&utm_medium=social.

T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., USA, 1 edition, 1997. ISBN 0070428077.

S. Moro, P. Cortez, and R. Laureano. Using data mining for bank direct marketing: An application of the crisp-dm methodology. 10 2011.

S. Moro, P. Cortez, and P. Rita. A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.*, 2014. doi: 10.1016/j.dss.2014.03.001.

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

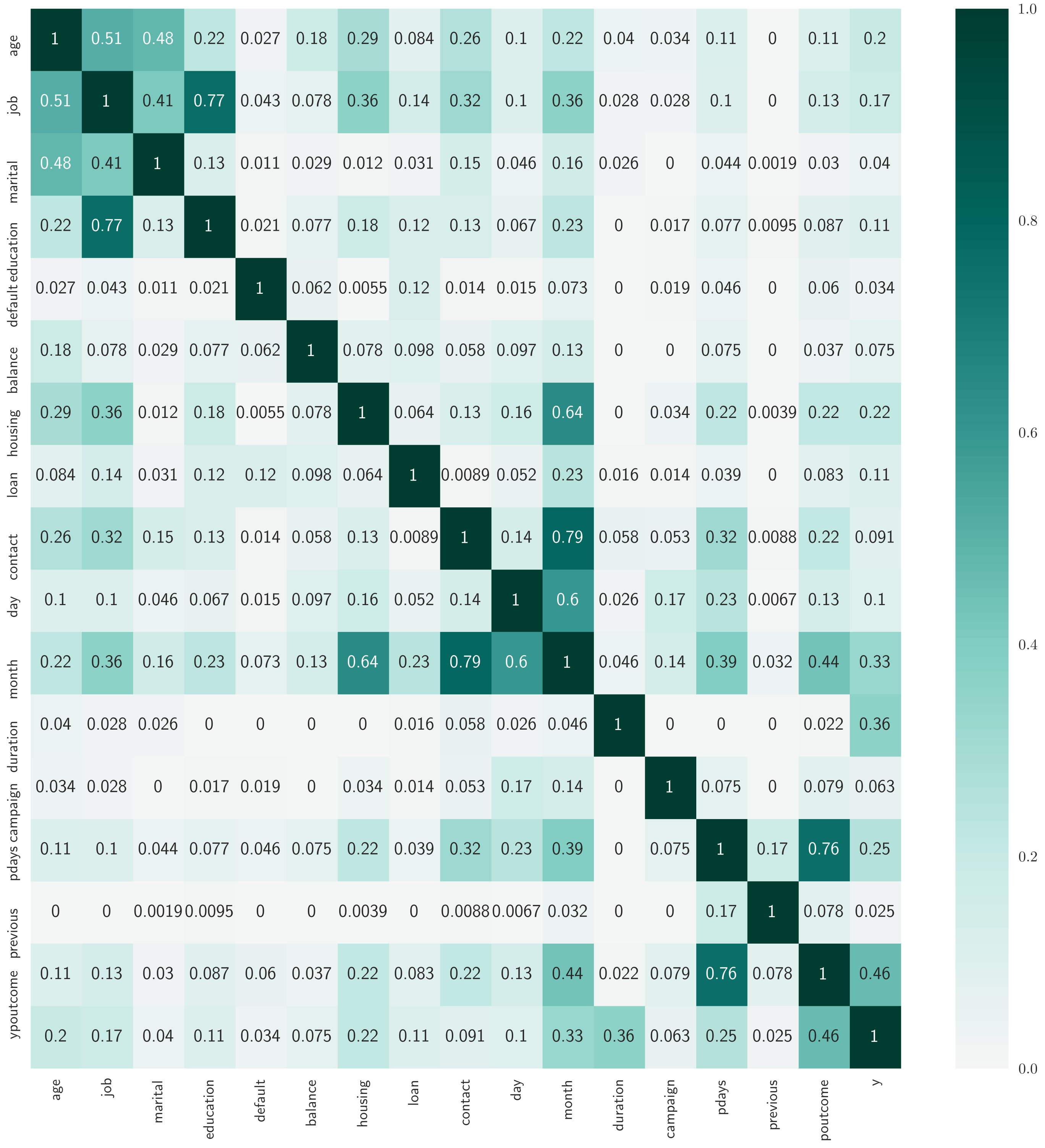
S. Raschka. Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning. *arXiv:1811.12808 [cs, stat]*, Nov. 2020. URL http://arxiv.org/abs/1811.12808. arXiv: 1811.12808.

G. Scott. Term Deposit Definition, Apr. 2021. URL https://www.investopedia.com/terms/t/termdeposit.asp.

E. tilo, O. lekan, A. ayi, and I. Ezekiel. Effect of Direct Market On Consumer Patronage In Selected Banks In Ondo State. *International Journal of Economics and Management Studies*, 7(8):77–82, Aug. 2020. ISSN 23939125. doi: 10.14445/23939125/IJEMS-V7I8P111. URL http://www.internationaljournalssrg.org/IJEMS/paper-details?Id=684.

K. M. Ting. Confusion Matrix. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 260–260. Springer US, Boston, MA, 2017. ISBN 978-1-4899-7687-1. doi: 10.1007/978-1-4899-7687-1_50. URL https://doi.org/10.1007/978-1-4899-7687-1_50.

A. Zheng and A. Casari. Feature Engineering for Machine Learning. In *Feature Engineering for Machine Learning*, pages 29–31. O'Reilly Media, Inc., Apr. 2018. URL https://www.oreilly.com/library/view/feature-engineering-for/9781491953235/.



	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
age	1	0.51	0.48	0.22	0.027	0.18	0.29	0.084	0.26	0.1	0.22	0.04	0.034	0.11	0	0.11	0.2
job	0.51	1	0.41	0.77	0.043	0.078	0.36	0.14	0.32	0.1	0.36	0.028	0.028	0.1	0	0.13	0.17
marital	0.48	0.41	1	0.13	0.011	0.029	0.012	0.031	0.15	0.046	0.16	0.026	0	0.044	0.0019	0.03	0.04
education	0.22	0.77	0.13	1	0.021	0.077	0.18	0.12	0.13	0.067	0.23	0	0.017	0.077	0.0095	0.087	0.11
default	0.027	0.043	0.011	0.021	1	0.062	0.0055	0.12	0.014	0.015	0.073	0	0.019	0.046	0	0.06	0.034
balance	0.18	0.078	0.029	0.077	0.062	1	0.078	0.098	0.058	0.097	0.13	0	0	0.075	0	0.037	0.075
housing	0.29	0.36	0.012	0.18	0.0055	0.078	1	0.064	0.13	0.16	0.64	0	0.034	0.22	0.0039	0.22	0.22
loan	0.084	0.14	0.031	0.12	0.12	0.098	0.064	1	0.0089	0.052	0.23	0.016	0.014	0.039	0	0.083	0.11
contact	0.26	0.32	0.15	0.13	0.014	0.058	0.13	0.0089	1	0.14	0.79	0.058	0.053	0.32	0.0088	0.22	0.091
day	0.1	0.1	0.046	0.067	0.015	0.097	0.16	0.052	0.14	1	0.6	0.026	0.17	0.23	0.0067	0.13	0.1
month	0.22	0.36	0.16	0.23	0.073	0.13	0.64	0.23	0.79	0.6	1	0.046	0.14	0.39	0.032	0.44	0.33
duration	0.04	0.028	0.026	0	0	0	0	0.016	0.058	0.026	0.046	1	0	0	0	0.022	0.36
campaign	0.034	0.028	0	0.017	0.019	0	0.034	0.014	0.053	0.17	0.14	0	1	0.075	0	0.079	0.063
pdays	0.11	0.1	0.044	0.077	0.046	0.075	0.22	0.039	0.32	0.23	0.39	0	0.075	1	0.17	0.76	0.25
previous	0	0	0.0019	0.0095	0	0	0.0039	0	0.0088	0.0067	0.032	0	0	0.17	1	0.078	0.025
poutcome	0.11	0.13	0.03	0.087	0.06	0.037	0.22	0.083	0.22	0.13	0.44	0.022	0.079	0.76	0.078	1	0.46
y	0.2	0.17	0.04	0.11	0.034	0.075	0.22	0.11	0.091	0.1	0.33	0.36	0.063	0.25	0.025	0.46	1

Figure 4: ϕ_k Correlation Heatmap

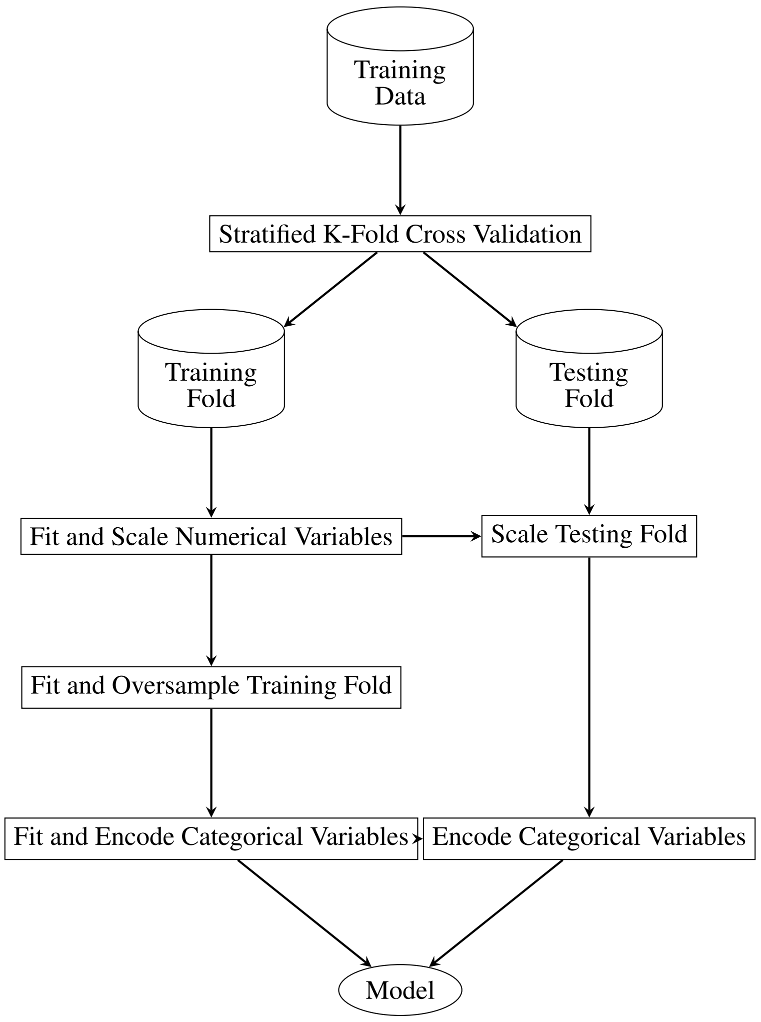

Training Data
Stratified K-Fold Cross Validation
Training Fold
Testing Fold
Fit and Scale Numerical Variables
Scale Testing Fold
Fit and Oversample Training Fold
Fit and Encode Categorical Variables
Encode Categorical Variables
Model

Figure 5: Data Pre-processing Flow