forecasting

MDPI

Article

A Generalized Flow for B2B Sales Predictive Modeling: An Azure Machine-Learning Approach

Alireza Rezazadeh

Electrical and Computer Engineering Department, University of Illinois at Chicago, Chicago, IL 60607, USA; arezaz2@uic.edu

Received: 3 July 2020; Accepted: 5 August 2020; Published: 6 August 2020

Abstract: Predicting the outcome of sales opportunities is a core part of successful business management. Conventionally, undertaking this prediction has relied mostly on subjective human evaluations in the process of sales decision-making. In this paper, we addressed the problem of forecasting the outcome of Business to Business (B2B) sales by proposing a thorough data-driven Machine-Learning (ML) workflow on a cloud-based computing platform: Microsoft Azure Machine-Learning Service (Azure ML). This workflow consists of two pipelines: (1) An ML pipeline to train probabilistic predictive models on the historical sales opportunities data. In this pipeline, data is enriched with an extensive feature enhancement step and then used to train an ensemble of ML classification models in parallel. (2) A prediction pipeline to use the trained ML model and infer the likelihood of winning new sales opportunities along with calculating optimal decision boundaries. The effectiveness of the proposed workflow was evaluated on a real sales dataset of a major global B2B consulting firm. Our results implied that decision-making based on the ML predictions is more accurate and brings a higher monetary value.

Keywords: costumer relation management; business to business sales prediction; machine learning; predictive modeling; microsoft azure machine-learning service

1. Introduction

In the Business to Business (B2B) commerce, companies compete to win high-valued sales opportunities to maximize their profitability. In this regard, a key factor for maintaining a successful B2B enterprise is the task of forecasting the outcome of sales opportunities. B2B sales process typically demands significant costs and resources and, hence, requires careful evaluations in the very early steps. Quantifying the likelihood of winning new sales opportunities is an important basis for appropriate resource allocation to avoid wasting resources and sustain the company's financial objectives [1–4].

Conventionally, forecasting the outcome of sales opportunities is carried out mostly relying on subjective human rating [5–8]. Most of the Customer Relationship Management (CRM) systems allow salespersons to manually assign a probability of winning for new sales opportunities [9]. This probability is then used at various stages of the sales pipeline, i.e., calculating a weighted revenue of the sales records [10,11]. Often each salesperson develops a non-systematic intuition to forecast the likelihood of winning a sales opportunity with little to no quantitative rationale, neglecting the complexity of the business dynamics [9]. Moreover, as often as not, sales opportunities are intentionally underrated to avoid any internal competition with other salespersons or overrated to circumvent the pressure from sales management to maintain a higher performance [12].

Even though the abundance of data and improvements in statistical and machine-learning (ML) techniques have led to significant enhancements in data-driven decision-making, the literature is scarce in the subject of B2B sales outcome forecasting. Yan et al. [12] explored predicting win-propensity of sales opportunities using a two-dimensional Hawkes dynamic clustering technique. Their approach allowed for live assessment of active sales although relied heavily on regular updates and inputs from salespersons in the CRM system. This solution is hard to maintain in larger B2B firms considering each salesperson often handles multiple opportunities in parallel and would put less effort into making frequent interaction with each sales record [13].

Tang et al. [9] built a sales forecast engine consist of multiple models trained on snapshots of historical data. Although their paradigm is focused on revenue forecasting, they demonstrated the effectiveness of hybrid models for sales predictive modeling. Bohane et al. [5] explored the idea of single and double-loop learning in B2B forecasting using ML models coupled with general explanation methods. Their main goal was actively involving users in the process of model development and testing. Built on their earlier work on effective feature selection [14] they concluded random forest models were the most promising for B2B sales forecasting.

Here, we proposed an end-to-end cloud-based workflow to forecast the outcome of B2B sales opportunities by reframing this problem into a binary classification framework. First, an ML pipeline extracts sales data and improves them through a comprehensive feature enhancement step. The ML pipeline optimally parameterizes a hybrid of probabilistic ML classification models trained on the enhanced sales data and eventually outputs a voting ensemble classifier. Second, a prediction pipeline makes use of the optimal ML model to forecast the likelihood of winning new sales opportunities. Importantly, the prediction pipeline also performs thorough statistical analysis on the historical sales data and specifies appropriate decision boundaries based on sales monetary value and industry segment. This helps to maximize the reliability of predictions by binding the interpretation of model results to the actual data.

The proposed workflow was implemented and deployed to a global B2B consulting firm's sales pipeline using Microsoft Azure Machine-Learning Service (Azure ML). Such a cloud-based solution readily integrates into the existing CRM systems within each enterprise and allows for more scalability. Finally, we compared the performance of the proposed solution with salespersons' predictions using standard statistical metrics (e.g., accuracy, AUC, etc.). To make the comparison more concrete, we also investigated the financial aspect of implementing this solution and compared the monetary value of our ML solution with salespersons' predictions. Overall, we have found that the proposed ML solution results in a superior prediction both in terms of statistical and financial evaluations; therefore, it would be a constructive complement to the predictions made by salespersons.

This paper is organized as follows: In Section 2, materials and methods used in this work are introduced in detail. Section 3 summarizes the results of this work. Section 4 presents discussion on the results, limitations of the current work and potential future directions.

2. Materials and Methods

2.1. Data and Features

Data for this study were obtained from a global multi-business B2B consulting firm's CRM database in three main business segments: Healthcare, Energy, and Financial Services (Finance for short). This section, first, gives an overview of the data and then explains a data enhancement technique used to infer additional relevant information from the dataset.

Data

A total number of 25,578 closed sales opportunity records starting January 2015 through August 2019 were used in this work (Figure 1a). Each closed opportunity record contained a status label (won/lost) corresponding to its ultimate outcome, otherwise if still active in the sales pipeline, they were labeled as open. Out of all closed sales records ∼58% were labeled as "won" in their final sales status (Figure 1b).

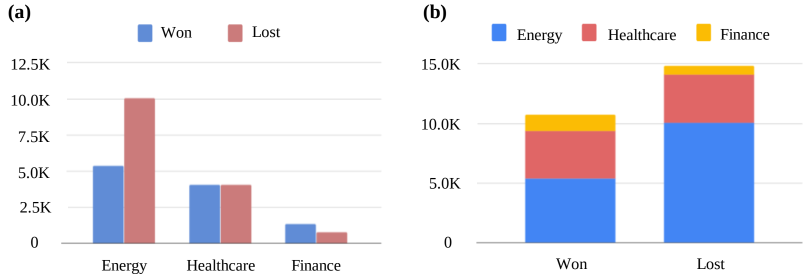


Figure 1. Data Exploration: (**a**) Distribution of sales opportunity records across the three business segments: Healthcare, Energy, and Finance. (**b**) Closed sales opportunities final status.

A total number of 20 relevant variables (features) were extracted for each sales opportunity from the raw CRM database. Table 1 describes these features in more details. Specifically, a subset of the features described the sales project (Opportunity Type, General Nature of Work, Detailed Nature of Work, Project Location, Project Duration, and Total Contract Value, Status). The remaining features provided further information on the customer (Account, Account Location, Key Account Energy, Key Account Finance, and Key Account Healthcare) and the internal project segmentation and resource allocation (Business Unit, Engagement Manager, Sales Lead, Probability, Sub-practice, Practice, Group Practice, Segment, and User-entered Probability).

Table 1. Raw CRM sales database features.

Count	Feature Type	Features
13	Categorical	Business Unit, Opportunity Type, Project Location, General Nature of Work, Detailed Nature of Work, Account, Account Location, Sales Lead Engagement Manager, Sub-practice, Practice, Group Practice, Segment
4	Binary	Status, Key Account Energy, Key Account Healthcare, Key Account Finance
3	Continuous	User-entered Probability, Project Duration, Total Contract Value

Once a sales opportunity profile was created in the CRM system, users were required to input their estimation for the probability of winning that opportunity. Please note that the user-entered probabilities were not used in the process of training ML models and were only used as a point of reference to compare with the performance of the ML workflow. All the features listed in Table 1 were required to populate in the CRM system; therefore, less than 1% of the dataset contained missing values. As a result, sales records with a missing value were dropped from the dataset.

2.2. Feature Enhancement

The CRM raw dataset was enhanced by inferring additional relevant features calculated across the sales records. These additional features were calculated using statistical analysis on the categorical features: Sales Leads, Account, Account Location, etc. Mainly, the idea was to extract a lookup table containing relevant statistics calculated across the sales records for each of the unique values in the categorical features.

By collecting the historical data of unique values of each categorical features (i.e., for each individual Sales Lead, Account, and Project Location, etc.), we calculated the following statistical metrics: (1) Total number of sales opportunities (2) Total number of won sales (3) Total number of lost sales (4) Average contract value (value for short) of won sales (5) Standard error of the mean won sales value (6) Winning rate calculated as the ratio of won and total sales counts (7) Coefficient of variation (the ratio of the standard deviation to the mean) [15] of won sales value to capture the extent of variability in the won sales contract values.

The aforementioned statistics were calculated and stored in feature enhancement lookup tables for each categorical feature (see Table 1 for a list of these features). Table 2 provides an example of a feature enhancement lookup table calculated based on the "Sales Lead" feature in the raw CRM dataset. These lookup Tables (13 tables in total for all categorical features) were appropriately merged back to the raw CRM sales dataset.

In the last feature enhancement step, the Mahalanobis [16] distance was calculated between each sales opportunity's value and the distribution of all won sales value that shared a similar categorical feature (individually for each of the 13 categorical features). This quantifies how far a sales value is relative to the family of won sales with the same characteristics (i.e., same Sales Lead, Project Location, Segment, etc.). The process of feature enhancement increased the total number of features to 137 for each sales record (20 features originally from the raw CRM dataset + $9 \times 13 = 117$ additional features from the lookup tables).

Table 2. Feature Enhancement Lookup Table: An example of the statistics calculated based on "Sales Lead" including counts of the total, won, and lost opportunities along with the mean and standard error of the mean (SEM) for won sales value and their coefficient of variation (CV).

	Sales Lead	Total	Won	Lost	Won Value Mean	Won Value SEM	Win Rate	CV
1	John Doe 1	3788	1902	1866	107,249.3	15,460.3	0.5	6.9
2	John Doe 2	1908	1908	0	3793.6	38.9	1.0	97.5
3	John Doe 3	1335	1232	103	5352.0	218.1	0.9	24.5
4	John Doe 4	986	454	566	492,626.8	90,913.8	0.5	5.4
5	John Doe 5	973	359	614	15,700.3	1283.0	0.4	12.2
...	...	...	...	...	...	...	...	...

The enhanced CRM dataset (25,578 total number of sales opportunities) was randomly split into a "train set" (70%) and a "test set" (30%). The train set was used to train ML models. The performance of the model on train set is reported using a 10-fold cross-validation technique. The test set was used to report the performance of the trained ML model on the unseen portion of the dataset. For further evaluations, after the proposed framework was deployed to the sales pipeline of the enterprise, a "validation set" was collected of new sales records over a period of 3 months (846 closed sales opportunities).

2.3. Machine-Learning Overview

Our solution to predicting the likelihood of winning sales opportunities is essentially reframing this problem in a supervised binary classification paradigm (won, lost). Hence, we made use of two of the most promising supervised classification algorithms: XGBoost, and LightGBM. In particular, these two models were selected among the family of probabilistic classification models due to their higher classification accuracy in our problem. A second motivation for using these two models was the fact that the distributed versions of both can easily integrate into cloud platforms such as Azure ML. Last, to attain a superior performance, multiple iterations of both models were combined in a voting ensemble.

2.3.1. Binary Classification

Probabilistic classification algorithms [17], given pairs of samples and their corresponding class labels $(X_1, y_1), \ldots, (X_n, y_n)$, capture a conditional probability distribution over the output classes $P(y_i \in Y \mid X_i)$ where for a binary classification scenario $Y \in \{0, 1\}$ (maps to lost/won in our problem). Given the predicted probability of a data sample, a decision boundary is required to define a reference point and predict which class the sample belongs to. In a standard binary classification, the predicted class is the one that has the highest probability [18]. This translates to a standard decision boundary of 0.5 for predicting class labels.

However, the decision boundary can be calibrated arbitrarily to reflect more on the distribution of the data. The influence of the decision boundary on the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) in binary classification is illustrated in Figure 2 (see Table 3 for definitions). In this work, we find the optimal decision boundary for a classification model by maximizing all true conditions (both TP and TN) which in return, minimizes all the false conditions (FP and FN). Visually, this decision boundary is a vertical line passing through the intersections of $P(X|Y=0)$ and $P(X|Y=1)$ in Figure 2.

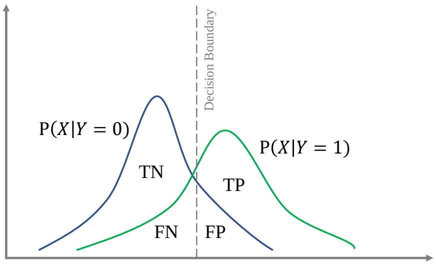


Figure 2. Decision Boundary: impact of the decision boundary on different scenarios of the binary classification.

The performance of a binary classifier can be evaluated using standard statistical metrics such as accuracy, precision, recall, and F1-score (see Table 3). For the case of binary classification, the area under ROC curve (AUC) measures the robustness of the classification (a higher AUC suggests more robust classification performance) [19]. As shown by Hand et al. [20], the AUC of a classifier G can be calculated as:

$$\hat{A} = \frac{S_0 - n_0(n_0+1)/2}{n_0 n_1} \tag{1}$$

where n_0 and n_1 are the numbers of positive and negative samples, respectively. Also, $S_0 = \sum r_i$, where r_i is the rank of the ith positive example in the ranked list where more positive examples are ranked higher.

We took a step forward to obtain more insight into the classification results and measured the performance of the classifier from a monetary aspect, i.e., we calculated the value created by adopting a classification algorithm in the decision-making process. In particular, we aggregated the total sales values in each of the four scenarios of classification (TP_m, TN_m, FP_m, FN_m) and defined monetary performance metrics with a similar formulation to the statistical metrics (see Table 3). For instance, the monetary precision is the fraction of the sales values correctly predicted as won.

Table 3. Statistical and monetary classifier performance metrics.

Statistical Metrics	
Notation	**Definition**
TP	True Positive: number of class 1 samples classified as 1
TN	True Negative: number of class 0 samples classified as 0
FP	False Positive: number of class 0 samples classified as 1
FN	False Negative: number of class 1 samples classified as 0
Metric	**Definition**
Precision	$TP/(TP+FP)$
Recall	$TP/(TP+FN)$
Accuracy	$(TP+TN)/(TP+TN+FP+FN)$
F1-Score	$2/(Recall^{-1}+Precision^{-1})$
Monetary Metrics	
Metric	**Definition**
$\mathrm{Precision_m}$	$TP_m/(TP_m+FP_m)$
$\mathrm{Recall_m}$	$TP_m/(TP_m+FN_m)$
$\mathrm{Accuracy_m}$	$(TP_m+TN_m)/(TP_m+TN_m+FP_m+FN_m)$

2.3.2. XGBoost and LightGBM Classifiers

XGBoost, introduced by Chen and Guestrin [21], is a supervised classification algorithm that iteratively combines weak base learners into a stronger learner. With this algorithm, the objective function J is defined as

$$J(\Theta) = L(y, \hat{y}) + \Omega(\Theta) \tag{2}$$

where Θ denotes the model's hyperparameters. The training loss function L quantifies the difference between the prediction $\hat{y}$ and actual target value y. The regularization term Ω penalizes the complexity of the model with the L1 norm to smooth the learned model and avoid over-fitting. The model's prediction is an ensemble of k decision trees from a space of trees $\mathcal{F}$:

$$\hat{y}_i = \sum_{i=1}^{k} f_k(x_i), f_k \in \mathcal{F} \tag{3}$$

The objective function at iteration t for n instances can be simplified as:

$$J^{(t)} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{t} \Omega(f_k) \tag{4}$$

where according to Equation (3), $\hat{y}_i$ can iteratively be written as

$$\hat{y}_i{}^{(t)} = \sum_{i=1}^{t} f_k(x_i) = \hat{y}_i{}^{(t-1)} + f_t(x_i) \tag{5}$$

The regularization term can be defined as

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{6}$$

where the coefficient γ is the complexity of each leaf. Also, T is the total number of leaves in the decision tree. To scale the weight penalization, λ can be tweaked. Using second-order Taylor expansion and assuming a mean square error (MSE) loss function, Equation (4) can be written as

$$J^{(t)} \approx \sum_{i=1}^{n}[g_i w_{q(x_i)} + \frac{1}{2}(h_i w_{q(x_i)})^2] + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{7}$$

Since each incident of data corresponds to only one leaf, according to [22], this can also be simplified as

$$J^{(t)} \approx \sum_{j=1}^{T}[(\sum_{i \in I_j} g_i) w_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \tag{8}$$

where I_j represents all instances of data in leaf j. As can be seen in Equation (8) minimizing the objective function can be transformed into finding the minimum of a quadratic function.

In an endeavor to reduce the computation time of the XGBoost algorithm, Ke, et al. proposed LightGBM [23]. The main difference between XGBoost and LightGBM is how they grow the decision tree (see Figure 3 for a high-level comparison). In XGBoost decision trees are grown horizontally (level-wise) while with LightGBM decision trees are grown vertically (leaf-wise). Importantly, this makes LightGBM an effective algorithm to handle datasets with high dimensionality.

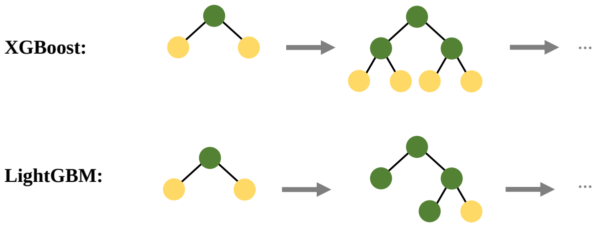


Figure 3. Comparison between XGBoost level-wise horizontal tree growth and LightGBM vertical leaf-wist tree growth.

2.3.3. Voting Ensemble

The main idea behind voting ensembles is to combine various classification models to balance out their individual classification errors. Voting ensembles predict the class label either by using most individual models' predictions (hard vote) [24] or averaging their predicted probabilities (soft vote) [25]. A voting ensemble was used to integrate the predictions of multiple iterations of both XGBoost and LightGBM classifiers with different parameterizations. Specifically, a soft-voting weighted average voting ensemble was used to combine the predictions for each model (Figure 4). A soft-voting ensemble is a meta-classifier model that computes its predicted probabilities y_i by takes the weighted average (w_j) probability predicted by each classifier (p_{ij}):

$$\hat{y}_i = \text{argmax}_i \sum_{j=1}^{m} w_j p_{ij} \tag{9}$$

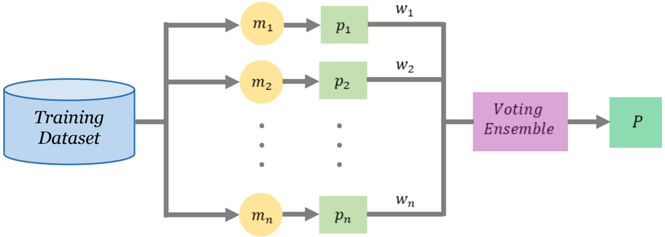


Figure 4. Voting Ensemble combines predictions p_i of multiple classifiers m_i using weighted average w_i to compute a final prediction P.

2.4. Workflow and Pipelines

Pipeline is defined as an executable workflow of data that is encapsulated in a series of steps. In this work, the proposed workflow consists of two main pipelines: (1) ML pipeline and (2) Prediction pipeline. All pipeline codes were custom-written in Python 3.7 on Microsoft Azure Machine-Learning Service [26] cloud platform. XGBoost v1.1 and LightGBM v2.3.1 libraries were integrated into Python to create ML classification models. The voting ensemble was created using the Microsoft Automated Machine-Learning tool [27].

2.4.1. Machine-Learning Pipeline

The main objective of the ML pipeline is to train predictive models on the closed sales opportunities data. As illustrated in Figure 5, there are four main steps in this pipeline:

(1) *Data Preparation*: Raw data of all closed sales opportunities are extracted from the CRM cloud database. Relevant features are selected for each sales record (see Table 1) and paired with their sales outcome (won/lost) as a class label. Please note that the user-entered probabilities are dropped to avoid biasing the model's predictions.

(2) *Feature Enhancement*: As described in Section 2.2, statistical analysis is performed on all categorical features to generate feature enhancement lookup tables for each of these categorical features (see Table 2). All lookup tables are stored back in the CRM cloud database. These tables are then appropriately merged back to the original selected features in the raw data.

(3) *Machine Learning*: A total number of 35 iterations of XGBoost and LightGBM classifiers with various parameterizations are trained on the data (Section 2.3.2). Eventually, all trained models are combined to generate a soft-voting ensemble classifier (Section 2.3.3).

(4) *Deploy Model to Cloud*: In the last step of the ML pipeline, the ensemble model is deployed as a web service using Azure ML. Azure ML platform supports creating a model's endpoint on Azure Kubernetes Service (AKS) cluster [28]. AKS enables request-response service with low latency and high scalability which makes it suitable for production-level deployments.

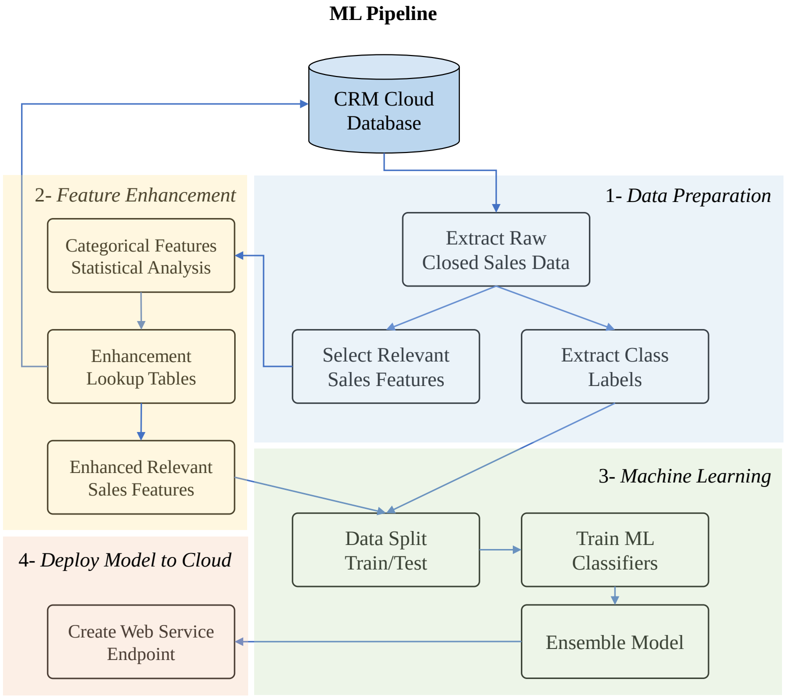


Figure 5. ML Pipeline: In four major steps the pipeline extracts and enhances sales data from a cloud database, trains an ensemble of ML classification models on the data, and eventually creates a cloud endpoint for the model.

2.4.2. Prediction Pipeline

The prediction pipeline, as illustrated in Figure 6, was designed to use the trained ML model and make predictions on the likelihood of winning new sales opportunities in four main steps:

(1) *Data Preparation*: All open sales records are extracted from the CRM cloud database. Relevant features are selected similar to the feature selection step in the ML pipeline. Please note that open sales opportunities are still active in the sales process and, hence, there is no final sales status (won/lost) assigned to them yet.

(2) *Feature Enhancement*: To make predictions on unseen data using the trained ML model, new input data needs to have a format similar to the data used to train the model. Therefore, all the previously stored lookup dictionaries are imported from the CRM cloud database and appropriately merged to the relevant features.

(3) *Machine-Learning Prediction*: The ensemble model created in the ML pipeline is called using its endpoint. The model makes predictions on the unseen sales data and assigns a probability of winning to each new opportunity.

(4) *Decision Boundaries*: All historical data on closed sales opportunities along with their ML predicted probabilities are grouped by the business segments (Healthcare, Energy, and Finance). Next, within each business segment, closed sales records are split into four quartiles based on their value. Then, the optimal decision boundary is calculated for each business segment's value quartile as described in Section 2.3.1. A total number of 12 decision boundaries are calculated (3 business segments × 4 quartiles). Eventually, all predicted probabilities and decision boundaries are stored back to the cloud database.

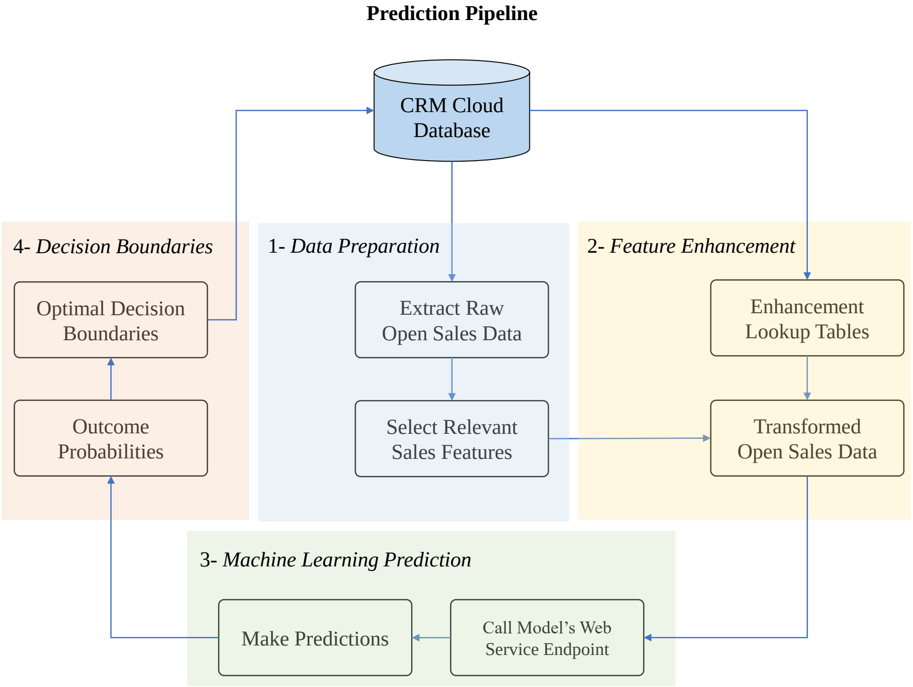

Prediction Pipeline
CRM Cloud Database
4- Decision Boundaries
Optimal Decision Boundaries
Outcome Probabilities
1- Data Preparation
Extract Raw Open Sales Data
Select Relevant Sales Features
2- Feature Enhancement
Enhancement Lookup Tables
Transformed Open Sales Data
3- Machine Learning Prediction
Make Predictions
Call Model's Web Service Endpoint

Figure 6. Prediction Pipeline: new sales opportunities data are transformed and enhanced, probability of winning is inferred using the trained ML model, and finally decision boundaries are optimized based on historical sales data.

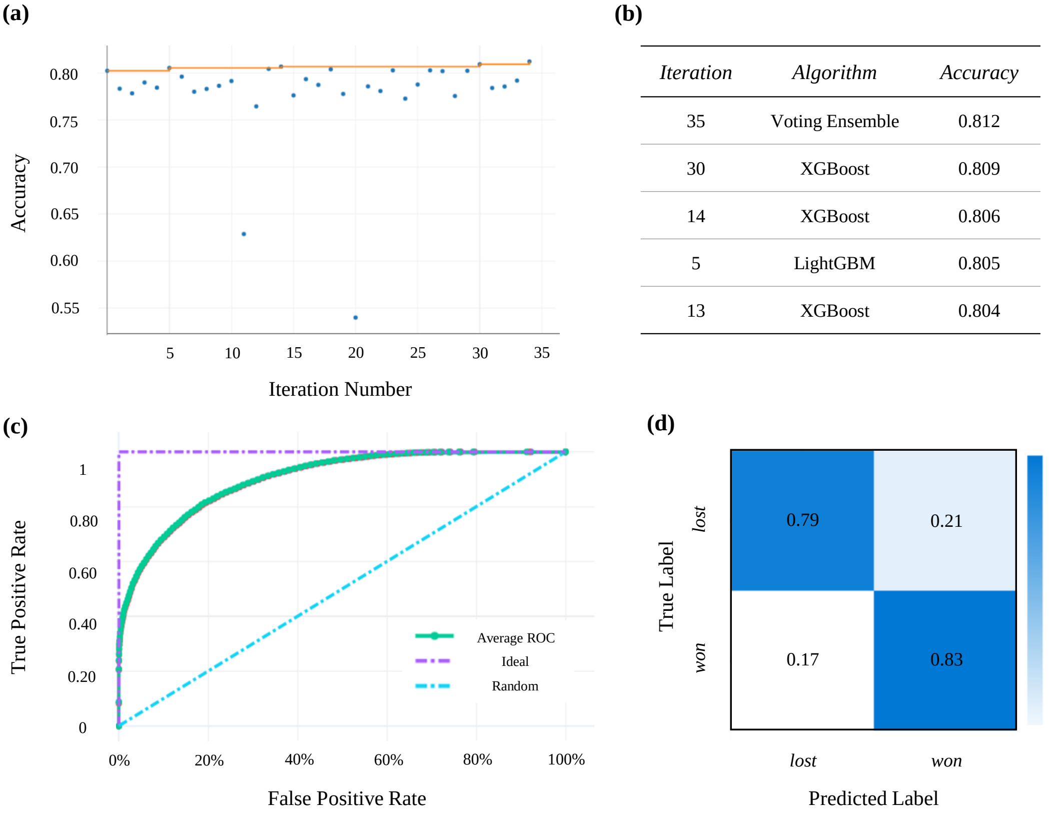
3. Results

This section gives an overview of the proposed workflow's performance. The workflow was implemented in the CRM system of a global B2B consulting firm. The two pipelines were scheduled for automated runs on a recurring basis on the Azure ML platform. The ML pipeline was scheduled for a weekly rerun to retrain ML models on updated sales data and generate updated feature enhancement lookup tables. The prediction pipeline was scheduled for a daily rerun to calculate and store predictions for new sales opportunities.

3.1. Training the ML Model

A total number of 34 iterations of XGBoost and LightGBM were individually trained on the data and then combined in a voting ensemble classifier (see Section 2.3.3 for more details). The training accuracy was calculated using 10-fold cross-validation. The accuracy for each of the 35 iterations (with the last iteration being the voting ensemble classifier) is demonstrates in Figure 7a. Training accuracy for the top five model iterations are listed in Figure 7b. As expected, the voting ensemble had a slightly higher training accuracy compared to each individual classifier.

The voting ensemble classifier had a training accuracy of 81% (other performance metrics are listed in Table 4). On the train set, approximately 83% of the won sales and 79% of the lost sales were classified correctly (Figure 7d). For more insight into the training performance ROC curve (Figure 7c) is also illustrated. The area under the ROC curve (AUC) was equal to 0.87. In other words, this implies that a randomly drawn sample out of the train set has a 87% chance of being correctly classified by the trained model.



Iteration	*Algorithm*	*Accuracy*
35	Voting Ensemble	0.812
30	XGBoost	0.809
14	XGBoost	0.806
5	LightGBM	0.805
13	XGBoost	0.804

Figure 7. ML Training Results: (**a**) Training accuracy for all model iteration. (**b**) Accuracy of the top five iterated models sorted by the training accuracy. (**c**) ROC curve of the voting ensemble classifier. (**d**) Confusion matrix showing the four scenarios of classification for the voting ensemble model.

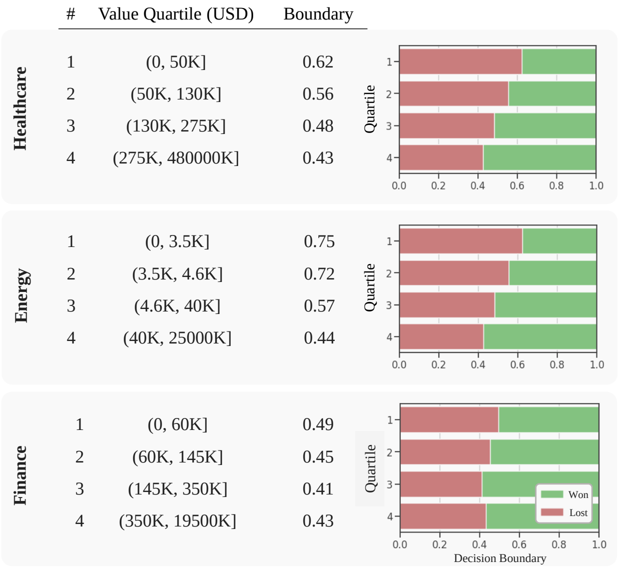
Table 4. Voting Ensemble Training Performance.

Metric	**Value**
Precision	0.81
Recall	0.83
Accuracy	0.82
AUC	0.87

3.2. Setting the Decision Boundaries

As explained in Section 2.4.2, statistical analysis of historical sales data is performed in each business segment (Healthcare, Energy, and Finance) to determine the decision boundaries. Specifically, the decision boundary was optimized for each of the four sales value quartiles of each business segment. The decision boundaries, demonstrated in Figure 8, ranged from 0.41 (Finance business segment—3rd value quartile) to 0.75 (Energy business segment—1st value quartile).

Interestingly, the decision boundaries were lower for sales opportunities with a higher monetary value which implies a more optimistic decision-making for more profitable opportunities. This sensible trend observed in the optimal decision boundaries provides more evidence to substantiate the idea of tailoring the boundaries uniquely to each business segment and value quartile due to their inherent decision-making differences.



	#	Value Quartile (USD)	Boundary
Healthcare	1	(0, 50K]	0.62
	2	(50K, 130K]	0.56
	3	(130K, 275K]	0.48
	4	(275K, 480000K]	0.43
Energy	1	(0, 3.5K]	0.75
	2	(3.5K, 4.6K]	0.72
	3	(4.6K, 40K]	0.57
	4	(40K, 25000K]	0.44
Finance	1	(0, 60K]	0.49
	2	(60K, 145K]	0.45
	3	(145K, 350K]	0.41
	4	(350K, 19500K]	0.43

Figure 8. Decision Boundaries.

3.3. Model's Performance

The voting ensemble was used to make predictions on the unseen test set. In particular, after inferring the probability of winning for each sales opportunity, they were classified in accordance with a decision boundary corresponding to their business segment and value quartile. If the inferred probability of winning exceeded the decision boundary, a sales opportunity was classified to be won otherwise it was classified to be a lost opportunity. To make a concrete comparison between user-entered and ML predictions, statistical and monetary performance metrics were calculated for both approaches.

All four classification scenarios in the test set for both user-entered and ML prediction are depicted in Figure 9a. Qualitatively, the ML workflow made fewer false classifications (i.e., compare the true positive TP slice proportions in Figure 9a). More specifically, the ML workflow accurately classified 87% of the unseen sales data while the user-entered predictions only had an accuracy of 67%. In fact, all statistical performance metrics (precision, recall, and F1 score) were in favor of the ML predictions (see Table 5).

The performance of the user-entered and ML predictions was also compared with reference to the monetary metrics (see Section 2.3.1 for more details). As shown in Figure 9b, sales opportunities falsely predicted to be won by the ML workflow had considerably lower cumulative monetary value (compare the true positive FP_m slice proportions). This implies a lower monetary loss due to prediction error when using the ML predictions. Quantitatively, the monetary accuracy of the ML model was notably higher than the user-entered (90% versus 74%). Other monetary performance metrics are listed in Table 5.

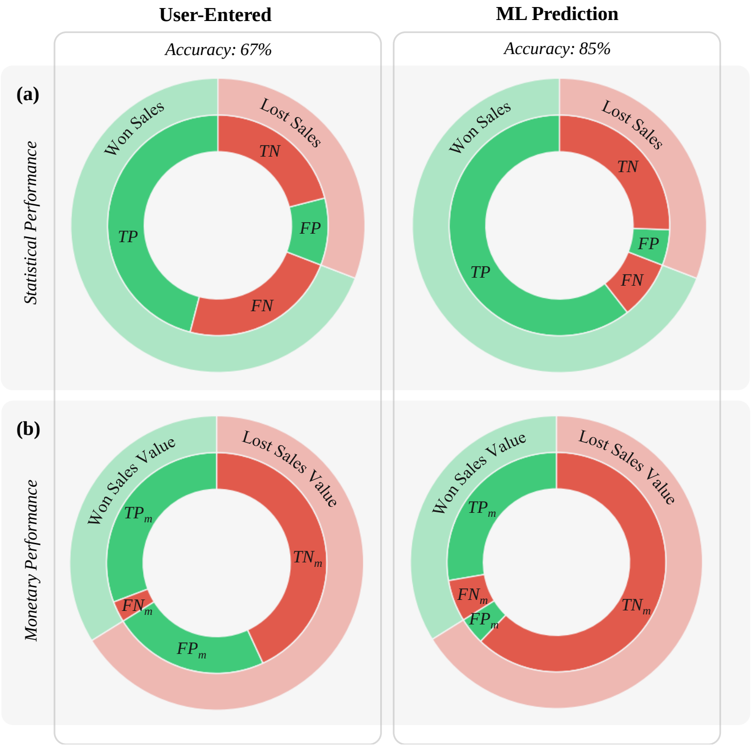


Figure 9. Test Set Results: (**a**) Statistical Performance: Actual outcome of all won (light green) and lost (light red) sales opportunities along with the corresponding predictions (solid green and red). (**b**) Monetary Performance: Cumulative contract value of won (light green) and lost (light red) sales opportunities along with the cumulative value of opportunities in each of the four classification scenarios (solid green and red). In both panels miss-matching colors indicates false classification.

Table 5. Test Set Performance Metrics.

Statistical Performance		
Metric	**User-Entered**	**ML**
Precision	0.82	0.92
Recall	0.66	0.87
F1-Score	0.73	0.89
Accuracy	0.67	0.85
Monetary Performance		
Metric	**User-Entered**	**ML**
Precision_m	0.57	0.87
Recall_m	0.91	0.82
Accuracy_m	0.74	0.90

3.4. Analysis of the Workflow Implementation

Similar to the previous section, a performance comparison between the user-entered and ML predictions was performed on a validation set. The validation set was collected while the workflow was implemented in the sales pipeline of a B2B consulting firm over a period of three months (see Section 2.1 for further details). A qualitative comparison in terms of statistical and monetary performance is presented in Figure 10. In the validation set, the ML workflow retained a substantially higher prediction accuracy (83% versus 63%). Also, there was an evident gap between the number of won sales misclassified by each approach (compare the true positive TP slices in Figure 10a).

The monetary accuracy of the ML predictions was marginally lower than the user-entered predictions (75% versus 77%). However, the cumulative value of the won sales opportunities correctly classified by the ML workflow was still considerably higher than the user-entered predictions (compare the true positive TP_m slices in Figure 10b). All performance metrics are listed in Table 6.

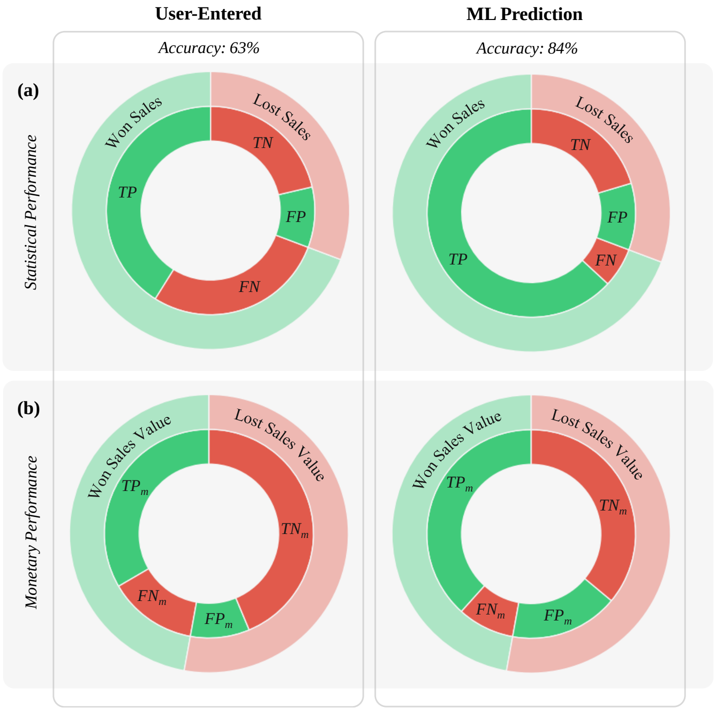


Figure 10. Validation Set Results: (**a**) Statistical and (**b**) Monetary performance of user-entered and ML predictions. Refer to Figure 9 caption for further explanations.

Table 6. Validation Set Performance Metrics.

Statistical Performance		
Metric	**User-Entered**	**ML**
Precision	0.82	0.85
Recall	0.59	0.92
F1-Score	0.70	0.87
Accuracy	0.63	0.83
Monetary Performance		
Metric	**User-Entered**	**ML**
Precision_m	0.79	0.71
Recall_m	0.72	0.82
Accuracy_m	0.77	0.75

4. Conclusions

In this paper, we proposed a novel ML workflow implemented on a cloud platform for predicting the likelihood of winning sales opportunities. With this approach, sales data were extracted from the CRM cloud database and then improved by an extensive feature enhancement approach. The data was then used to train an ensemble of probabilistic classification models in parallel. The resulting meta classification model was then used to infer the likelihood of winning new sales opportunities. Lastly, to maximize the interpretability of the predictions, optimal decision boundaries were calculated by performing statistical analysis on the historical sales data.

To inspect the effectiveness of the ML approach, it was deployed to a multi-business B2B consulting firm for over three months. The performance of the ML workflow was compared with the user-entered predictions made by salespersons. Standard statistical performance metrics confirmed that by far the ML workflow provided superior predictions. From a monetary standpoint, the value created from decision-making was also higher when incorporating the ML workflow.

The proposed ML workflow is a cloud-based solution that can readily be integrated into the existing cloud-based CRM system of enterprises. On top of that, this workflow is highly sustainable and scalable since it relies on cloud computing power instead of on-premise computing resources. Although our proposed workflow is mainly built around Azure ML platform, future work can explore implementing this workflow on other cloud computing resources such as Amazon web services, Google cloud platform, etc.

A potential issue with the proposed workflow is handling the scenario of imbalanced dataset. An imbalanced dataset is characterized by having more instances of a certain class compared to others [29]. In our problem, this would translate to a dataset that has more lost sales record than won (or vice versa). For instance, consider Energy or Finance business segments (Figure 1) in our data set where the number of won and lost sales records are unbalanced. Solutions to deal with an imbalance problem at the data-level involves oversampling the smaller class or undersampling the larger class [30–32]. Galar et al. [33] showed that combining random undersampling techniques with ensemble models stands out among other data-level solutions. In future work, we hope to explore this idea by incorporating an undersampling technique to the existing ensemble models in the workflow.

The results obtained in this work suggest a data-driven ML solution for predicting the outcome of sales opportunities is a more concrete and accurate approach compared to salespersons' subjective predictions. However, it is worth mentioning that ML solutions should not be overwhelmingly used to rule out sensible or justifiable sentiments of salespersons in evaluating a sales opportunity. A data-driven approach, such as the workflow presented in this work, can provide a reliable reference point for further human assessments of the feasibility of a sales opportunity.

Funding: This research received no external funding.

Acknowledgments: The author would like to thank Saeed Fotovat for his valuable suggestions and comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Monat, J.P. Industrial sales lead conversion modeling. *Mark. Intell. Plan.* 2011. Available online: https://www.emerald.com/insight/content/doi/10.1108/02634501111117610/full/html (accessed on 5 August 2020). [CrossRef]
2. Bohanec, M.; Borštnar, M.K.; Robnik-Šikonja, M. Integration of machine learning insights into organizational learning: A case of B2B sales forecasting. In *Blurring the Boundaries through Digital Innovation*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 71–85.
3. Matthies, B.; Coners, A. Double-loop learning in project environments: An implementation approach. *Expert Syst. Appl.* **2018**, *96*, 330–346. [CrossRef]
4. Duran, R.E. Probabilistic sales forecasting for small and medium-size business operations. In *Soft Computing Applications in Business*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 129–146.
5. Bohanec, M.; Robnik-Šikonja, M.; Borštnar, M.K. Organizational learning supported by machine learning models coupled with general explanation methods: A Case of B2B sales forecasting. *Organizacija* **2017**, *50*, 217–233. [CrossRef]
6. Ingram, T.N.; LaForge, R.W.; Schwepker, C.H.; Williams, M.R. *Sales Management: Analysis and Decision Making*; Routledge: Los Angeles, CA, USA, 2015.
7. Davis, D.F.; Mentzer, J.T. Organizational factors in sales forecasting management. *Int. J. Forecast.* **2007**, *23*, 475–495. [CrossRef]
8. Armstrong, J.S.; Green, K.C.; Graefe, A. Golden rule of forecasting: Be conservative. *J. Bus. Res.* **2015**, *68*, 1717–1731. [CrossRef]

9. Xu, X.; Tang, L.; Rangan, V. Hitting your number or not? A robust & intelligent sales forecast system. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 3613–3622.
10. Davis, J.; Fusfeld, A.; Scriven, E.; Tritle, G. Determining a project's probability of success. *Res. Technol. Manag.* **2001**, *44*, 51–57. [CrossRef]
11. De Oliveira, M.G.; Rozenfeld, H.; Phaal, R.; Probert, D. Decision making at the front end of innovation: The hidden influence of knowledge and decision criteria. *R D Manag.* **2015**, *45*, 161–180. [CrossRef]
12. Yan, J.; Gong, M.; Sun, C.; Huang, J.; Chu, S.M. Sales pipeline win propensity prediction: A regression approach. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 854–857.
13. Lambert, M. Sales Forecasting: Machine Learning Solution to B2B Sales Opportunity Win-Propensity Computation. Ph.D. Thesis, National College of Ireland, Dublin, Ireland, 2018.
14. Bohanec, M.; Kljajić Borštnar, M.; Robnik-Šikonja, M. Feature subset selection for B2B sales forecasting. In Proceedings of the 13th International Symposium on Operational Research, SOR, Bled, Slovenia, 23–25 September 2015; Volume 15, pp. 285–290.
15. Abdi, H. Coefficient of variation. *Encycl. Res. Des.* **2010**, *1*, 169–171.
16. De Maesschalck, R.; Jouan-Rimbaud, D.; Massart, D.L. The mahalanobis distance. *Chemom. Intell. Lab. Syst.* **2000**, *50*, 1–18. [CrossRef]
17. Abbott, D. Foreword 2 for 1st Edition. In *Handbook of Statistical Analysis and Data Mining Applications*; Academic Press: Cambridge, MA, USA, 2018; pp. xv–xvi.
18. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
19. Bewick, V.; Cheek, L.; Ball, J. Statistics review 13: Receiver operating characteristic curves. *Crit. Care* **2004**, *8*, 1–5.
20. Hand, D.J.; Till, R.J. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Mach. Learn.* **2001**, *45*, 171–186.
21. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
22. Zhang, D.; Qian, L.; Mao, B.; Huang, C.; Huang, B.; Si, Y. A data-driven design for fault detection of wind turbines using random forests and XGboost. *IEEE Access* **2018**, *6*, 21020–21031. [CrossRef]
23. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 3146–3154.
24. Ruta, D.; Gabrys, B. Classifier selection for majority voting. *Inf. Fusion* **2005**, *6*, 63–81. [CrossRef]
25. Kuncheva, L.I.; Rodríguez, J.J. A weighted voting framework for classifiers ensembles. *Knowl. Inf. Syst.* **2014**, *38*, 259–275. [CrossRef]
26. Barga, R.; Fontama, V.; Tok, W.H. Introducing microsoft azure machine learning. In *Predictive Analytics with Microsoft Azure Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 21–43.
27. Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.; Blum, M.; Hutter, F. Efficient and robust automated machine learning. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 2962–2970.
28. Barnes, J. Azure machine learning. In *Microsoft Azure Essentials*, 1st ed.; Microsoft Press: Redmond, WA, USA, 2015.
29. Sun, Y.; Wong, A.K.; Kamel, M.S. Classification of imbalanced data: A review. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 687–719. [CrossRef]
30. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357.
31. Chawla, N.V.; Japkowicz, N.; Kotcz, A. Special issue on learning from imbalanced data sets. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 1–6. [CrossRef]

32. Zhou, Z.H.; Liu, X.Y. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. Knowl. Data Eng.* **2005**, *18*, 63–77. [CrossRef]
33. Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **2011**, *42*, 463–484. [CrossRef]