universität
wien

BACHELORARBEIT / BACHELOR’S THESIS

Titel der Bachelorarbeit / Title of the Bachelor’s Thesis

„Predicting Sales Pipeline for CRM Data Using Machine Learning Methods“

verfasst von / submitted by

Natalia Tretiakova

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Bachelor of Science

Wien, 2020 / Vienna, 2020

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 033 521
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Bachelor Computer Science
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Inform.Univ. Dr.Claudia Plant

Contents

1 Acknowledgements 3

2 Motivation 3

2.1 Problem definition . . . 3

2.2 Current approach . . . 4

2.3 Drawbacks of current approach . . . 4

2.4 Goals and metrics . . . 5

2.5 Work process and methods . . . 5

2.6 Summary . . . 6

3 Related Work 6

4 Major Idea 8

4.1 Proposal approach . . . 8

4.1.1 1. Step: Classification Model . . . 9

4.1.2 2. Step: Regression Model . . . 9

4.1.3 3. Step: Calculate probable monthly/quarterly revenue . . . 10

4.2 Benefits . . . 10

5 Implementation 11

5.1 Data structure . . . 12

5.2 Data gathering . . . 13

5.2.1 Generated data . . . 13

5.2.2 Real data . . . 13

5.3 Data preprocessing and separation . . . 14

5.4 Models . . . 15

5.4.1 Models used for classification problem (first step) . . . 15

5.4.2 Models used for regression problem (second step) . . . 16

5.5 Making predictions . . . 17

5.6 Accuracy metrics . . . 18

5.6.1 1. Step: Classification . . . 18

5.6.2 2. Step: Regression . . . 18

5.6.3 3. Step: Predicted Revenue . . . 18

5.7 Code compilation . . . 18

6 Evaluation and Discussion 19

6.1 Evaluation methods . . . 19

6.2 Evaluation results . . . 20

6.2.1 Evaluation on generated data . . . 20

6.2.2 Evaluation on real data . . . 23

6.3 Discussion . . . 24

7 Conclusions and Future Work **25**
7.1 Summary 25
7.2 Limitations 26
7.3 Future work 26

Glossary **29**

Acronyms **29**

Appendices **31**

A Performance summary for generated data **31**
A.1 Neural Networks 31
A.2 Logistic Regression 33

B Performance summary for real data **34**
B.1 Neural Networks 34
B.2 Logistic Regression 36

Abstract

Though forecasting of sales performance in Business-To-Business (b2b) market is a very important task, the methods used still base mostly on subjective estimations of sales managers. Whereas Machine Learning (ml) methods are widely used for revenue forecasting in Business-To-Consumer (b2c) area, in Customer Relationship Management (crm) they has not been used yet due to complexity of the data and the forecasting task. In this thesis an attempt of using ml methods for predicting sales pipeline performance in b2b area will be presented and compared to sophisticated methods.

1 Acknowledgements

This thesis was written in collaboration with *smartPM.solutions* - a provider of CPM (Corporate Performance Management) and SPM (Sales Performance Management) solutions for customers all over the world. The internal sales pipeline data was kindly provided by its Chief Executive Officer (ceo) Alexander Hein. The insights about existing problems and sales management process were given by Sebastian Wallner and Marina Hein.

Though this thesis focuses on one particular company, the sales pipeline approach is widely used. I believe that the other companies face the similar problems as *SmartPM Solutions*. Hence I assume that the results of this research could bring benefit not only to this company, but also to its clients.

2 Motivation

2.1 Problem definition

Many companies which are specialized on the sales of Software Solutions and Consulting in B2b area have a problem, that the sales process of their services may extend to a relatively long time period, while the size of an offer is usually quite large. From the first contact with the potential customer up to actual sales date usually some month are passed, while the size of the Deal (also often called Opportunity) may vary from tens to hundreds thousands of Euro. These characteristics of b2b transactions make the planning and predictions of cash flow to a struggling task for ceo. The existing solutions lead frequently to very inaccurate turnover forecasts.

Another problem is that a lot of time is being invested into the Opportunities which are then being rejected from the customer after very long period simply because the sales managers overestimate the likelihood of success. An earlier detection of the sales Opportunities with smaller success probability could help to change the sales and communication strategy with the particular customer or just not further consider the Opportunity for revenue planning.

Some authors have already noticed, that though b2c sales forecasting is meanwhile a very well-researched subject, b2b area still remains mostly undiscovered [17, 9]. Some researchers, among them Brynjolfsson [10], claim that Data-Driven Decision making (ddd) increases the company productivity and revenue output for 5-6%, which means that using of statistical and ml methods in b2b area is worth investing time and effort. In this paper I will analyse the current approach for estimating sales probability for a deal and will try to find a better solution for revenue forecasting using ml algorithms like Logistic Regression (logreg), Neural Networks (nn) and Random Forest Regression (rf).

Stage	**Description**	**Probability, %**
1. Marketing	Opportunity is identified and information is recorded	5
2. Prospect	Intensive contact with potential buyer	10
3. Discovery	Understanding the client's needs and buying position	25
4. Identify Pains	Understanding the client's problems	50
5. Value Proposition	Proposing a solution	80
6. Final Bid	Defining a concrete solution and budget	90
Won	Opportunity closed and won	100
Lost	Opportunity closed and lost	0

Table 1: Sales pipeline stages

2.2 Current approach

The sales Opportunities usually go through several stages, each of them is being associated with a particular winning probability. There is no strict separation on stages, and basically every company is free to define their own stages. Though the basic idea is the same, the names and descriptions, as well as closing probability of every stage may vary [12]. In Internet there are plenty of sources which define from four [16] to seven [21, 22] Stages. In *SmartPM Solutions* the six-stages system is used with winning probabilities and stage definitions defined in Table 1.

Each of these stages is being assigned not only a corresponding probability of sales success, but also an estimation of closing date (the date when the contract will be signed). For example, in the first stage (Marketing) the success probability is set to 5% and closing date is set to some date, say, in 6 month. In the sixth stage "Final bid" the closing probability rises up to 90% and the closing date is being set probably to the next two weeks, or as made up with the customer. The turnover forecast for such pipelines is being calculated simply by multiplying the deal size with its closing probability as shown in equation 1 [7]. The result is the predicted revenue on the date which was estimated as probable closing date.

$$PredictedRevenue = DealSize * WinProbability \tag{1}$$

2.3 Drawbacks of current approach

The existing procedure has some problems. First, the estimated closing dates are being set mostly arbitrary or by the intuition of the Sales Manager and often not being updated until the quarter end is coming [19, 11]. The Sales Executive of *SmartPM Solutions* Sebastian Wallner partially confirms these assumptions:

> "I set the estimated close date after the first conversation with the customer and rely mostly on my experience and the intuition. In case when the estimated close date is coming closer and I know that it can not be hold, I simply shift it three months later. It is difficult to say how accurate my first estimations are, because I do not follow how often the sales deadlines have been shifted."

Second problem is that the success probabilities are in most cases the same for all customers and opportunities and are being set only based on the current pipeline stage of the opportunity. There usually exist a possibility in CRM systems to change the probabilities manually, but only few sales managers want to waste their time for estimating and updating them. Even if a sales manager sets these probabilities manually, these estimations are basically subjective, biased or even adjusted on what CEO wishes to see in the end of the quarter [11, 25].

The third problem is that Sales Managers spend their time filling the tables with estimated close dates and probabilities instead of focusing on more relevant tasks like communication with the customers. Often, as mentioned above, these metrics are being set and changed arbitrary and hence may be very inaccurate. Furthermore not all the Opportunities are being updated in time or documented properly [11] (we will face this problem later). The Sales Manager may not know exactly which factors are most important for success or failure of the particular Opportunity and therefore estimate the probabilities incorrectly.

2.4 Goals and metrics

The main goal of this work is to enable *automatic* estimation of winning probabilities and closing dates of the Opportunities, and hence better revenue forecasting, by using ML algorithms. By automatising of the winning probabilities estimation, the sales managers could win their time and concentrate on their customers rather than on filling the tables. But the primary goal of this thesis is to make better revenue forecasts, which would help the CEO to better estimate their cash flow.

Since the problem is rather complex and needs to be solved in several steps, which will be discussed in the section 4.1, the success metrics will include three parts:

1. Accuracy of win probability prediction (classification problem)
2. Accuracy of closing date estimation (regression problem)
3. Accuracy of quarterly revenue prediction

The algorithm will be tested on manually generated data as well as on real data with all its drawbacks, to simulate the process as close to the real situation as possible.

The predicted results for winning probability will be compared to guessed probabilities as well as to real outcome to see if ML may outperform the subjective estimation of winning probability and opportunity closing date. In the end the probable revenue will be calculated by using the guessed and predicted probabilities and estimated closing date as described in chapter 2.2 and compared to the real revenue. In addition, for the ML predictions the revenue will be calculated by taking the unweighted deal size of the opportunities which were classified as Won (see equation 2).

$$PredictedRevenue = \begin{cases} DealSize & if \;\; WinProbability \geq 0.5 \\ 0 & if \;\; WinProbability < 0.5 \end{cases} \tag{2}$$

2.5 Work process and methods

When I first was asked by *SmartPM Solutions* CEO Alexander Hein to find the solution how to make our revenue forecasts more accurate, I considered it as an uncomplicated problem. The scientific research showed though, that sales forecasting in B2B area has not got any sophisticated solution yet. The few papers about this task which I have found confirm that this area has not been thoroughly explored so far, in comparison to B2C sales forecasting [25, 17, 19]. In addition to that I have noticed, that the data is usually not being maintained well enough, which makes such a complex problem even more challenging.

For the deeper understanding of the problem and its possible solution scientific research together with the research in Web, as well as the interviews with sales manager and marketing team of *SmartPM Solutions* were taken into account. The communication with our sales executive Sebastian Wallner helped me to understand the current approach and its drawbacks, and to develop some understanding for dirty and messy data. After I noticed that none of the papers which I have found solve our particular problem (to estimate the sales revenue), but rather only concentrate on the estimation of closing probability, I decided to use the two-step approach described below in section 4.1.

The algorithm was tested on generated and real data. I have tested three different models for the first classification step (NN, LOGREG and Support Vector Classifier (SVC)) and two models for the second step (NN and RF). Due to very slow performance of SVC and no better accuracy, it was decided not to concern this model further. Also NN were not satisfying for the regression step, because it often gave only one value for all opportunities, thus only RF was considered. The NN and LOGREG models for the first step were then tested with various configurations to be able to estimate the best parameters combination. Each step and further the estimation of probable revenue was evaluated with appropriate accuracy metrics.

2.6 Summary

The approach proved to be more effective than sophisticated methods for both generated and real data. In both cases NN model performed better than LOGREG. The prediction of closing dates worked much better for generated data, but also for real data was significantly better than subjective estimations.

One interesting observation was that both predicted and guessed win probabilities and closing dates estimations for the real data were so bad that even the models which have predicted that none of the opportunities will be won, performed better in terms of Mean Absolute Error (MAE) for quarterly revenue and accuracy. Of course this is not a solution to a problem and those models should not be considered, therefore a combination of metrics was used for finding the best model.

Nevertheless, the accuracy of the closing probability predictions outperformed the subjective estimations significantly. The MAE of closing date prediction was better than initial guesses of sales manager, but the error was still very large. Therefore the estimations of quarterly revenues were respectively inaccurate. My suspicion is that the real data may probably handled so sloppy that some deals were not marked as closed on time and therefore the proper training was impossible.

All things considered I may summarize that ML methods may be arguably applied for the problems of this class, especially when the sales managers did not use to handle the data properly.

The thesis is structured as following: In section 3 the similar ideas will be discussed. My idea and proposal approach will be presented in section 4. Implementation details inclusive data description, models and metrics used can be found in section 5. Finally, in section 6 results will be discussed and evaluated.

3 Related Work

One possibility to make a revenue forecast in B2B area is to use aggregated sales volumes and formulate a problem as a regression task, which may be solved using Linear Regression (LINREG) or NN as for example in the paper of Luxhøj at al [18]. Though for my CEO it was rather interesting to predict the development of the existing pipeline and individual Opportunities in particular.

Fortunately, some researchers have faced this problem already and provided some solution ideas. Here I will give an overview of the most interesting of them.

D'haen and Van der Poel in their paper [12] make predictions for Leads which of them are going to become an Opportunity using Nearest Neighbour algorithm for the first phase and then LOGREG, Decision Tree (DT) and NN models in the second phse. A similar algorithm could be applied on Opportunities for predicting the most promising ones, like which Opportunity is going to become a closed Deal. However, the approach is not useful for predicting the date of closing, since this facet is not considered in the paper.

Femina Bahari and Sudheep Elayidom [6] face the problem from another point of view and choose the most promising and valuable customers using NN and Bayes classification models. Again, the researchers only predicted if the customer will accept the offer, but not the time frame when it most probably happens.

Junchi Yan et al. are using Hawkes Process model for predicting the next pipeline stage for each Opportunity and claim that this method outperforms the subjective ratings and some machine learning methods like Logistic and Cox model [25]. In the other paper [24] the same authors use LOGREG to estimate if an Opportunity will be won in a given time frame. Equivalently, this model predicts the status of an opportunity better than the estimations of the sales manager. The works of these researchers are corresponding very close to my task, since they consider not only the output classification, but also a time frame. The main difference is that the authors predict the outcome for the defined time frame (e.g. if an Opportunity will be won in the next three months), while my approach does not consider any time restrictions and the outcome is grouped by quarter in the last stage.

Eitle and Buxmann [13] use RF, SVC and XGBoost to predict the success of Opportunity and achieve an accuracy of about 80%, but as most of the researchers they do not predict the time when the success/failure event should happen, which is of main interest for my thesis.

Bohanec et al. in their papers "Explaining machine learning models in sales predictions" [8] and "Organizational Learning Supported by Machine Learning Models Coupled with General Explanation Methods: A Case of B2B Sales Forecasting" [9] combine the human and artificial intelligence for making the decisions in B2B market by computing the win probability predictions using various black-box ML methods such as NN and explaining them. The final decision is made by the sales experts. The explanation of decisions made by machine sounds very interesting and would be definitely part of my future work. The authors claim, that the decisions supported by ML helped to improve the sales opportunities forecasts. Analogously, their approach does not include the estimation of closing date though.

The approach of Gong et al. [15] is probably the nearest to mine: the researchers used LOGREG model to predict the closing probability for the Opportunities in defined time windows (quarterly). The authors mention, that they used only LOGREG model due to its simplicity and efficiency. The difference to my work is, that they first define the time window, when the predictions are of interest. In my approach I make the the predictions for all opportunities and then estimate the closing date with the regression model. Similarly to my models, their model outperforms the estimations of the seller significantly.

Monat [19] uses the scoring method for ranking the importances of features and calculates the win probability using these metrics. This is an interesting approach, but is not scalable, since in different domains the features may have various importances. Also the accuracy of the model was about 65%, which is not better that subjective sales manager estimations in my data.

Gentsch [14] in his book gives as an example of using Artifical Intelligence (AI) in sales by making predictions with DT and NN models and then ranking the most promising Leads of Opportunities. Unfortunately, he does not provide any implementation details, but rather gives an idea of how ML may be used for solving problems in this domain.

Additionally it is worth to mention that `Microsoft` presented a new feature called *Predictive Opportunity Scoring*[1], which uses ML models to estimate winning probabilities of Opportunities. At the moment of writing of this thesis my company switched to another product and therefore unfortunately there was no possibility for me to test this feature. Though, the fact that such giants as `Microsoft` are working on this topic shows that the prediction of opportunities outcome in B2B market is a known problem which gains on interest.

4 Major Idea

The major idea of my approach is to implement a solution, which allows to make **exact, automated** predictions for sales revenue, based on existing Opportunities. The preprocessing work has been minimized and mostly carried out by the machine, reducing the work of Sales Manager to minimum. Consequently, the aim of this approach is to exempt the sales manager from the responsibility to manually set the estimated closing dates and win probabilities of Opportunities, as well as to improve the predictions quality.

To achieve this goal, I decided to separate the problem into three parts: the first task is to predict which Opportunities will be won or lost (or in particular to calculate their successful closing probabilities). After that the closing date is being estimated. Then based on these two parameters as well as the volume of Opportunity the total revenue can be calculated as described in equations 1 and 2.

Since the data is being updated weekly, as it will be explained in the section 5.1, the idea is to find an algorithm which enables the possibility of creating a new forecast automatically every week based on the most up-to-date data. This means that data preparation and cleaning process is very limited and the algorithm should work basically on dirty data. The interaction with the data scientist after the process has been started should be minimized. This intention requires some grade of responsibility of the sales manager, who should at least correctly update the current state of the Opportunities.

4.1 Proposal approach

The approach of this thesis is to compute the winning probabilities and the expected close date of the Opportunities on each stage and time period using AI and ML methods such as NN, LOGREG and RF. With the help of NN or LOGREG classification model first the winning probability of a Deal is computed. Then based on the output of the classification model the probable closing dates are computed using regression model. In this case RF proved itself as the best choice. Later the probable revenue on the estimated closing dates is being calculated with two methods: using binary output of the first step or classification probabilities. The result can be then grouped by months or by quarters as it usually being made for the reports. The schema of the approach is shown on figure 1.

The aim of this approach is to make the revenue predictions based on the existing Opportunities in the pipeline rather than solving a regression problem by looking at the revenues in the previous

[1] `https://docs.microsoft.com/en-us/business-applications-release-notes/october18/ai/predictive-opportunity-scoring`

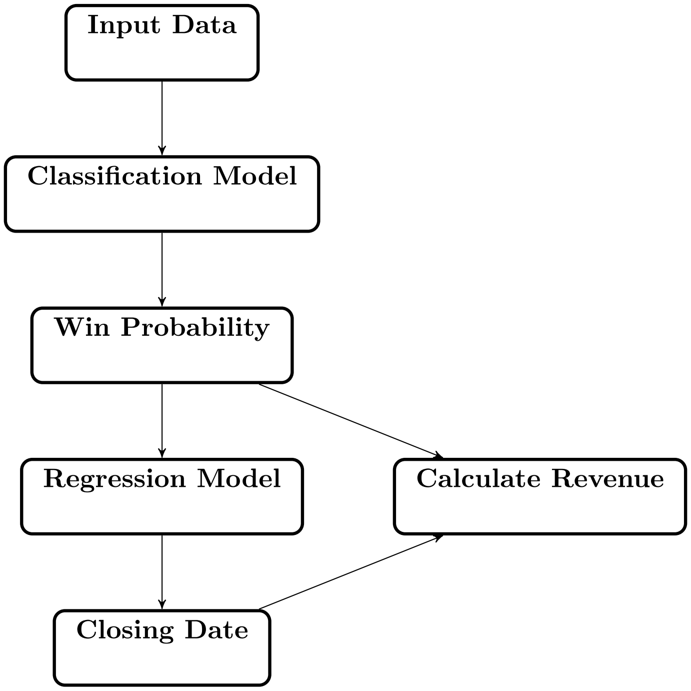


Figure 1: Suggested approach

periods. That means, the revenue predictions are made only based on the Deals existing in the current pipeline and do not consider the ones which may come later. Below each step is explained more specifically.

4.1.1 1. Step: Classification Model

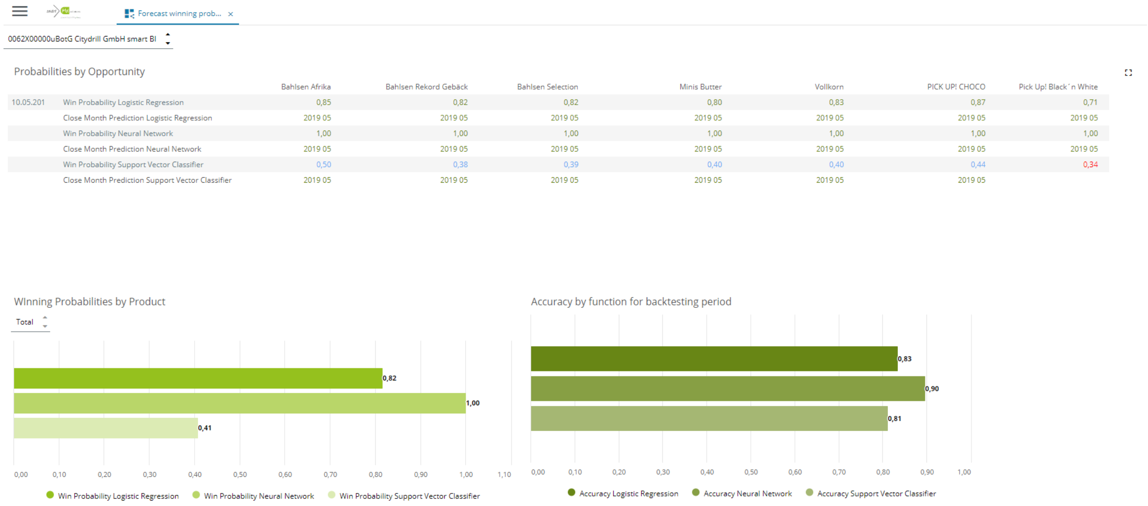
First, for each Deal its likelihood of being successfully closed is calculated using classification models. In this step initially three models were tested: NN, LOGREG and SVC. Since SVC was much slower and the accuracy was the worst, it was later omitted. The training and predictions were made using all features except number of days till closing, because this feature is "not known" yet, means can not be known for open Opportunities. Respectively this data was deleted from the training set and saved for the next step. The model gives as output the classification probabilities and classification results - both of them will be used later. The probabilities of Opportunities to be classified *"Won"* are used for the further revenue calculation and the binary classification result is utilized in the next step.

4.1.2 2. Step: Regression Model

The training data from the first step inclusive its output is now used for the estimation of closing dates. In this step we try to predict the number of days which will pass till the deal is closed by solving a regression problem. This number can not be negative an can only have integer values.

After converting and cleaning the output, the results are being added to the saved upload date to get the date of closing.

In Figure 2 the probable results of the first two stages are visualized.



0062X00000uBotG Citydrill GmbH smart BI

Probabilities by Opportunity

		Bahlsen Afrika	Bahlsen Rekord Gebäck	Bahlsen Selection	Minis Butter	Vollkorn	PICK UP! CHOCO	Pick Up! Black´n White
10.05.201	Win Probability Logistic Regression	0,85	0,82	0,82	0,80	0,83	0,87	0,71
	Close Month Prediction Logistic Regression	2019 05	2019 05	2019 05	2019 05	2019 05	2019 05	2019 05
	Win Probability Neural Network	1,00	1,00	1,00	1,00	1,00	1,00	1,00
	Close Month Prediction Neural Network	2019 05	2019 05	2019 05	2019 05	2019 05	2019 05	2019 05
	Win Probability Support Vector Classifier	0,50	0,38	0,39	0,40	0,40	0,44	0,34
	Close Month Prediction Support Vector Classifier	2019 05	2019 05	2019 05	2019 05	2019 05	2019 05	

WInning Probabilities by Product
Total
0,82
1,00
0,41
0,00
0,10
0,20
0,30
0,40
0,50
0,60
0,70
0,80
0,90
1,00
1,10
● Win Probability Logistic Regression
● Win Probability Neural Network
● Win Probability Support Vector Classifier
Accuracy by function for backtesting period
0,83
0,90
0,81
0,00
0,10
0,20
0,30
0,40
0,50
0,60
0,70
0,80
0,90
1,00
● Accuracy Logistic Regression
● Accuracy Neural Network
● Accuracy Support Vector Classifier

Figure 2: Estimated closing dates and winning probabilities for opportunities

4.1.3 3. Step: Calculate probable monthly/quarterly revenue

After the winning probabilities and closing dates have been estimated, the probable revenue can be calculated. This has been done using weighted (1) and unweighted (2) revenue calculation. Both methods will be evaluated in section 6.

In both cases the revenue is calculated for each opportunity and is set to the probable closing date which equals the date for which the forecast is calculated plus number of days till closing, estimated from step 2. Then these outcomes are being grouped by month and by quarter. For accuracy metrics only quarterly turnover estimations will be considered.

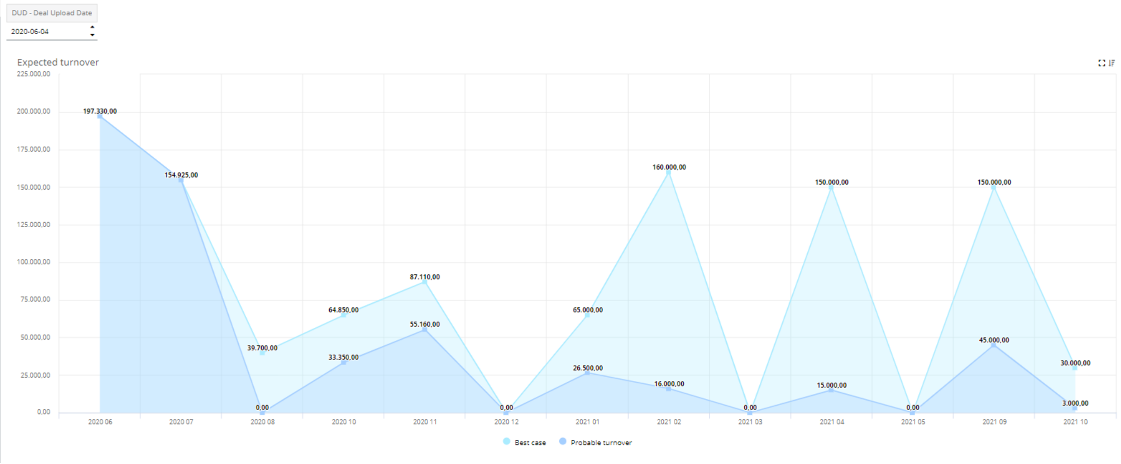
The example of visualisation is given in figure 3. Here one can choose the upload date for which the forecast was calculated and see the differences between the revenue estimations. In this case the light blue area shows the best case when all deals will be closed, and the dark blue area is the forecast made using weighted approach with predicted closing probabilities.

4.2 Benefits

The use of machine learning methods in sales pipeline performance forecast is expected to bring benefits for both sales managers and for the CEO.

1. For sales managers:

    - Saving the time which may be used for more interesting and relevant tasks. Currently the sales managers spend quite a lot of time with filling the `Excel` or similar tables



DUD - Deal Upload Date

2020-06-04

Figure 3: Example of visualisation of probable revenue estimation (deal size multiplied with sales probability) vs. best case (all deals will be closed)

with current stages, winning probabilities and estimated closing dates. These metrics are often very subjective and don't correspond with real situation.

- Sales manager may change the strategy for communication with a customer if she sees that the model predicts lower winning probabilities as expected. In the most helpless cases one may just not further concentrate on the particular Deal and switch to more promising ones.

2. For CEO:

   - Better estimation of sales revenue: when estimated close dates are being set arbitrary, the company is calculating with the potential revenue on the wrong time point. The more accurate estimation of closing dates can help in getting more reliable sales performance forecasts.
   - Artificial intelligence is able to consider multiple external factors and estimate which of them or which combination is most important for sales performance. Knowing these factors can help the CEO to focus on the weak points.

5 Implementation

The approach was implemented using mainly `Python` 3.7 [23] and in particular `Sklearn` library [20]. The data was generated in `R` which has only historical reasons and basically could in like manner be generated in `Python`. In the following section I will explain the data structure and preparation step and show how the data was gathered and generated. Besides that, the models and accuracy metrics used will be described. Finally the compilation details for code reproduction will be provided.

5.1 Data structure

Usually all the data regarding the Opportunities is being stored in some CRM system. It contains the name of the Opportunity, name of the client, some details about the client like the region or industry to which it belongs, name of corresponding sales manager, the associated products and their prices, the creation date and of course the current stage. The overview of the features being collected in *SmartPM Solutions* are listed in the table 2.

To enable analysis and changes tracking, the data is being exported weekly into `Unit4 Prevero` system with the corresponding upload date and also saved in the database. The anonymized example of how it then looks is presented in figure 4. It is made primarily for the ability to compare the Deals status on different time periods and to create the up-to-date reports and dashboards. I will use these weekly time-series data for each Opportunity to train the models and predict the closing stage for open Deals.

Feature	**Description**
Upload Date	Upload date
Created	Opportunity creation Date
Opportunity name	Usually the name of customer and the product group of interest
Industry	Industry group of the customer (Services, Retail, Finance etc.)
Region	Customer country
Marketing Campaign	From which marketing campaign the customer was gained
Customer Type	New or existing customer
Owner	Opportunity owner (name of sales manager)
Stage	Current stage
Product	Name of product
Price	Price of product
Quantity	Quantity of products
Value	$Price * Quantity$
Estimated Close Date	Expected closing date set by sales manager

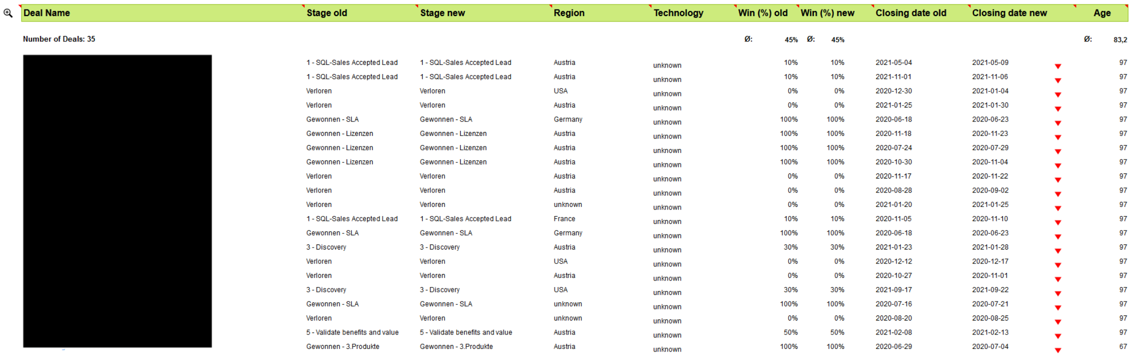
Table 2: Features description

The presented approach is meant to make the predictions after every data export. Some additional features are being derived from the data, these include:

- Last Stage - The stage the opportunity had last week
- Age - Number of days passed from opportunity creation till export date
- Expected Closing Days - Number of days from upload date to estimated close date

Additionally, two target columns for both classification and regression task were derived:

- Future Stage - If the opportunity will be won, lost or remain open
- Time to close - How many days will pass till closing



Deal Name	Stage old	Stage new	Region	Technology	Win (%) old	Win (%) new	Closing date old	Closing date new	Age
Number of Deals: 35					Ø: 45%	Ø: 45%			Ø: 83,2
[redacted]	1 - SQL-Sales Accepted Lead	1 - SQL-Sales Accepted Lead	Austria	unknown	10%	10%	2021-05-04	2021-05-09 ▼	97
[redacted]	1 - SQL-Sales Accepted Lead	1 - SQL-Sales Accepted Lead	Austria	unknown	10%	10%	2021-11-01	2021-11-06 ▼	97
[redacted]	Verloren	Verloren	USA	unknown	0%	0%	2020-12-30	2021-01-04 ▼	97
[redacted]	Verloren	Verloren	Austria	unknown	0%	0%	2021-01-25	2021-01-30 ▼	97
[redacted]	Gewonnen - SLA	Gewonnen - SLA	Germany	unknown	100%	100%	2020-06-18	2020-06-23 ▼	97
[redacted]	Gewonnen - Lizenzen	Gewonnen - Lizenzen	Austria	unknown	100%	100%	2020-11-18	2020-11-23 ▼	97
[redacted]	Gewonnen - Lizenzen	Gewonnen - Lizenzen	Austria	unknown	100%	100%	2020-07-24	2020-07-29 ▼	97
[redacted]	Gewonnen - Lizenzen	Gewonnen - Lizenzen	Austria	unknown	100%	100%	2020-10-30	2020-11-04 ▼	97
[redacted]	Verloren	Verloren	Austria	unknown	0%	0%	2020-11-17	2020-11-22 ▼	97
[redacted]	Verloren	Verloren	Austria	unknown	0%	0%	2020-08-28	2020-09-02 ▼	97
[redacted]	Verloren	Verloren	unknown	unknown	0%	0%	2021-01-20	2021-01-25 ▼	97
[redacted]	1 - SQL-Sales Accepted Lead	1 - SQL-Sales Accepted Lead	France	unknown	10%	10%	2020-11-05	2020-11-10 ▼	97
[redacted]	Gewonnen - SLA	Gewonnen - SLA	Germany	unknown	100%	100%	2020-06-18	2020-06-23 ▼	97
[redacted]	3 - Discovery	3 - Discovery	Austria	unknown	30%	30%	2021-01-23	2021-01-28 ▼	97
[redacted]	Verloren	Verloren	USA	unknown	0%	0%	2020-12-12	2020-12-17 ▼	97
[redacted]	Verloren	Verloren	Austria	unknown	0%	0%	2020-10-27	2020-11-01 ▼	97
[redacted]	3 - Discovery	3 - Discovery	USA	unknown	30%	30%	2021-09-17	2021-09-22 ▼	97
[redacted]	Gewonnen - SLA	Gewonnen - SLA	unknown	unknown	100%	100%	2020-07-16	2020-07-21 ▼	97
[redacted]	Verloren	Verloren	unknown	unknown	0%	0%	2020-08-20	2020-08-25 ▼	97
[redacted]	5 - Validate benefits and value	5 - Validate benefits and value	Austria	unknown	50%	50%	2021-02-08	2021-02-13 ▼	97
[redacted]	Gewonnen - 3.Produkte	Gewonnen - 3.Produkte	Austria	unknown	100%	100%	2020-06-29	2020-07-04 ▼	67

Figure 4: Anonymized example data from Unit4 Prevero CRM

5.2 Data gathering

The algorithm was tested on both generated and real data. Considering the messiness and poor quality of the real data, it was decided to test the algorithm also on "perfectly handled" generated data.

5.2.1 Generated data

The data was generated using `R` programming language and the approach was following:
First the lists of features were created and each value was assigned some unevenly distributed probability. Then a data frame was created where each opportunity had some number of products and respective region, industry and other features, inclusive stage. For each stage an estimated closing date was assigned, as it usually happens in real CRM table filling process. For example, the opportunities which have the stage "1. Marketing" got estimated close date in 22 weeks, the deals in the final stage got the estimated closing date in only four weeks. Finally the weekly updates were generated: each opportunity can jump into the next stage or remain in old stage with a certain probability. This probability depends on current stage and three other features: owner, industry and region. There are some best, worst and normal cases depending on which owner, industry or region the opportunity has. The best cases are most likely to be sold quickly and the worst cases would be most likely lost. Also the earlier stages usually last longer.

As the last step, the duplicated closed entries were deleted from the data set. The distribution of number of opportunities in each stage over the periods is shown in figure 5.

The data has 55632 rows with 1200 opportunities, 700 customers, each 10 regions and industries, 6 marketing campaigns, 15 different owners and 12 products. The periods range from 01-01-2017 to 01-01-2020. The total number of rows may wary, since the creation dates and number of products of each opportunity are being set randomly.

5.2.2 Real data

The real data was kindly provided by *SmartPM Solutions* CEO Alexander Hein. The prepossessing was minimal, as the aim was to keep the data as "real" as possible. The only manipulations were the anonymizing of the features, renaming of some stages, as well as dropping of the entries which

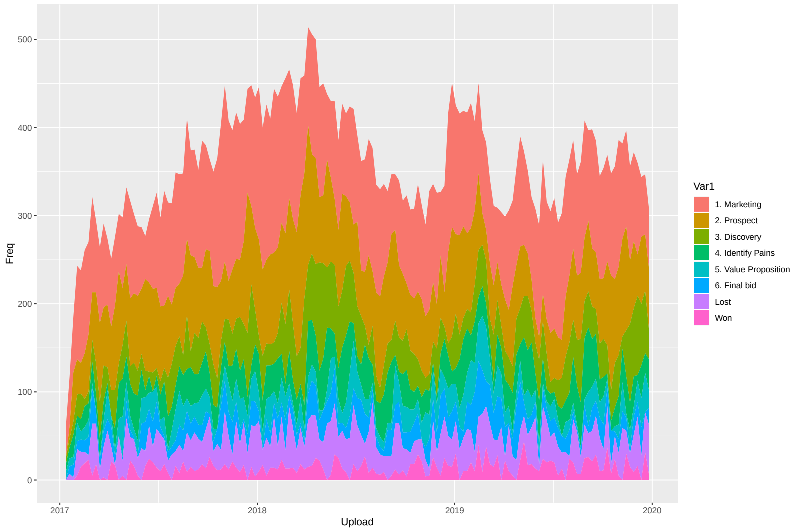


Figure 5: Distribution of stages over periods for generated data

were never opened or never closed, since we can not use them for training or testing. For privacy reasons the raw data will not be provided in the attachment.

After this cleaning and preparing process the real data had 69960 rows with 628 opportunities, 9 industries, 13 regions, 579 customers, 24 marketing campaigns (one of them covers 87% of all opportunities, which is obviously a result of bad data handling), 132 owners and 25 products. The upload dates range from 26-01-2018 to 10-05-2019, but creation dates range is smaller and has values from 03-01-2015 to 04-05-2019. One may see, that the structure of the data is slightly different from the generated data in some points, but it should not affect the results that much, since the idea is common.

Of course, some better data cleaning could probably lead to much better results, since some features were not handled properly. I consciously refused that hence the main idea was to enable automatic weekly predictions which would not require any manual data exploration and cleaning.

5.3 Data preprocessing and separation

As briefly mentioned above, for testing and training only the opportunities with known future closing stage were taken, that means only those, of which it is known if they will be lost or won. So the first preparation step was to determine the future stage of an opportunity and to drop those rows which were never closed. After determining the future stage and closing date, all the rows with current closed stage were dropped and only the open deals were used for training and testing.

Another important step was converting the string data types to numerical to be able to train models. For the classification step the non-numerical features were additionally preprocessed using One-Hot-Encoding (OHE). As for the regression task the Tree model was used, it worked better without OHE. In both cases the data was scaled using `MinMaxScaler` from `Sklearn` library.

Both generated and real data sets were separated into train and test sets. This separation was made by extracting opportunity names from the open data. Then the 30% of these names were taken for testing and remaining 70% for training. This is the only way to separate the data properly, since one opportunity usually has several uploads, which happen weekly. Therefore a simple separation in test-train set by taking random n% of rows for training would lead to cheating.

5.4 Models

Considering the two types of problems which should be solved, classification and regression, different ML models were used. For the first part NN and LOGREG models were taken into account. For the second part RF was the best choice, though initially also LINREG and NN were tested, but the results were not satisfying. In all cases the implementation of `Sklearn` was used, due to the fact that I have worked with this library a lot during my study and work and have already become quite familiar with it.

5.4.1 Models used for classification problem (first step)

Neural Networks. The class `MLPClassifier` from `Sklearn` library implements a multi-layer perceptron (see figure 6) algorithm that trains using Backpropagation [2].

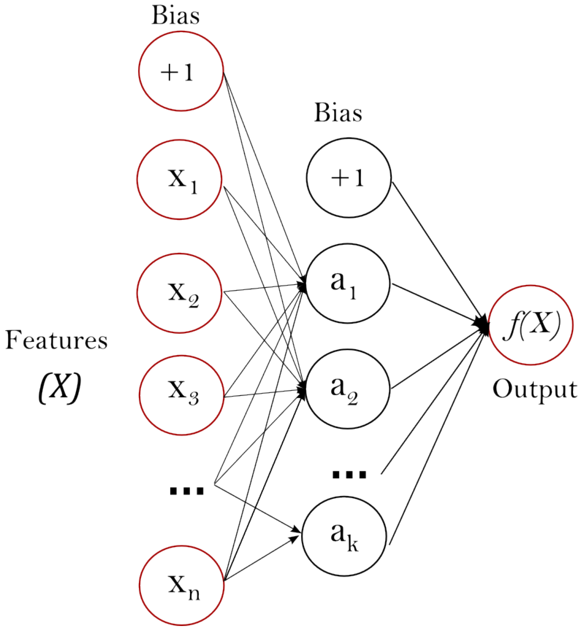


Figure 6: Multi-Layer Perceptron [2]

The NN model was evaluated with various parameter settings for activation function, solver and number of hidden layers and their nodes. Also some other parameters were different from default, as summarized in table 3. Another parameters were taken by default and are shown in listing 1.

Parameter	Values
Activation function	identity, logistic, tanh, relu
Solver	lbfgs, sgd, adam
Number of nodes and layers	(2, 2, 2), (100, 100), (20, 16, 10, 4), (100, 80, 60, 40)
Learning Rate	adaptive
Early stopping	True
Max iter	1000

Table 3: Parameter settings for NN model

```
class sklearn.neural_network.MLPClassifier(hidden_layer_sizes=(100, ),
    activation='relu', *, solver='adam', alpha=0.0001, batch_size='auto'
    , learning_rate='constant', learning_rate_init=0.001, power_t=0.5,
    max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=
    False, warm_start=False, momentum=0.9, nesterovs_momentum=True,
    early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2
    =0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000)
```

Listing 1: Default parameters for MLPClassifier [5]

Logistic Regression is a linear model for classification. Similarly to NN model, this was also trained with different parameter settings as summarized in table 4. The descriptions of all parameters may be found in the documentation [4] and the default parameter are as shown in listing 2.

Parameter	Values
Solver	newton-cg, lbfgs, liblinear, sag, saga
C	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
Class weight	balanced

Table 4: Parameter settings for LOGREG model

5.4.2 Models used for regression problem (second step)

Random Forest. In random forests each tree in the ensemble is built from a sample drawn with replacement from the training set [1].

For random forests only two different configurations were used: one for generated data and one for real data. The models were trained only once on the training sets before training the classification models and then recycled for the predictions on the test sets with various predictions made in the first step. The comparison of parameters id shown in table 5 and the default parameters are presented in listing 3.

```
class sklearn.linear_model.LogisticRegression(penalty='l2', *, dual=
    False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1,
    class_weight=None, random_state=None, solver='lbfgs', max_iter=100,
    multi_class='auto', verbose=0, warm_start=False, n_jobs=None,
    l1_ratio=None)
```

Listing 2: Default parameters for Logistic Regression [4]

Parameter	**Generated Data**	**Real Data**
n_estimators	100	100
criterion	mae	mae
n_jobs	-1	-1
min_samples_leaf	0.02	0.01
min_samples_split	0.01	0.3
max_samples	None	0.8

Table 5: Parameter settings for RF model

5.5 Making predictions

At first, the training for the regression step is made using the training data with known real future stage (won or lost). Then the whole range of various NN and LOGREG models are trained on the same data, but with the future stage as target column. After classification predictions are made, they are being attached to the test data on the place of future stage and the predictions for regression model are being produced. Consequently, the output of the first stage influences the performance of the regression stage.

After both predictions are ready, the revenue can be calculated. For the calculations two ways were used presented in sections 2.2 and 2.4. Given that the data for every opportunity represents a weekly time series, for the calculation of revenue the test data was iterated by its upload date. That means for each upload date the predicted revenue is calculated and compared to actual revenue separately. The results are then grouped by month and by quarter.

```
class sklearn.ensemble.RandomForestRegressor(n_estimators=100, *,
    criterion='mse', max_depth=None, min_samples_split=2,
    min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto
    ', max_leaf_nodes=None, min_impurity_decrease=0.0,
    min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=
    None, random_state=None, verbose=0, warm_start=False, ccp_alpha=0.0,
     max_samples=None)
```

Listing 3: Default parameters for Random Forest [3]

5.6 Accuracy metrics

The metrics for calculating the accuracy were chosen appropriately to the problem solved. All metrics were compared to real values along with the values estimated by sales manager. The "guessed" probabilities for the classification task were derived from the sales stages presented in table 1. The "guessed" numbers for regression task were calculated from the data during the preprocessing step by subtracting upload date form the estimated close date.

5.6.1 1. Step: Classification

For the classification problem a number of metrics was evaluated: precision, recall, f-1 score, accuracy, confusion matrix and Receiver Operating Characteristic Curve (ROC). The best model was picked by Area Under the Curve (AUC) value, where the best model should have the value closest to 1. This particular metric was chosen due to the manner of calculation the final forecast, where the probabilities are being multiplied with the deal amount. Another metrics were considered for the model comparison, but did not have influenced the decision for picking the best model as will be described in detail in section 6.

5.6.2 2. Step: Regression

As the accuracy metric for the regression task MAE (3) was considered as the most appropriate. This metric was not observed for the best model selection, bearing in mind that regression model configuration is constant for all classification models. The outcome of this step influences in some degree the prediction for the next step, however itself depends on the classification results.

$$MAE(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \hat{y_1}| \tag{3}$$

5.6.3 3. Step: Predicted Revenue

For this part equivalently to the regression task the MAE measure was used. To make a fair evaluation of the predictions, first the opportunities which were presented in the test data on the chosen upload date were first extracted from the entire real data. Then the outcome on actual closing dates for these opportunities was calculated and finally compared to the predicted revenue on the predicted closing dates. The results were then grouped by months and by quarters. The best model was chosen by the smallest quarterly MAE.

5.7 Code compilation

The code can be found by the following link: `https://github.com/tcubaruba/bachelor-project`

Installation & Preparations I used `pip` as packet management system and provided a `requirements.txt` file to easily install all necessary libraries.

Download the files with:

```
git clone https://github.com/tcubaruba/bachelor-project.git
```

Install pip with:

```
sudo apt update && sudo apt install python3-pip
```

Navigate to the directory PythonScripts with:

```
cd bachelor-project/Python_Scripts/
```

Install Python requirements with:

```
pip3  install -r requirements.txt
```

Running the code Navigate to **src** folder:

```
cd src
```

Run main.py:

```
python3 main.py
```

The program was tested on MacBook Pro with 2.3 GHz Quad-Core Intel Core i5 with 16GB RAM on MacOS 10.15.5 using Python 3.7.4. Random state was always set to 42 to make the results reproducible. The plots can be found in bachelor-project/Plots and the console output in bachelor-project/Outputs.

6 Evaluation and Discussion

6.1 Evaluation methods

The combination of three accuracy measures described in section 5.6 is important, since a superior performance in one task does not necessarily mean that the revenue predictions in the final analysis will be satisfying. For example, if the AUC value is close to 1, but the MAE for closing dates is very high, the revenue forecast will still be very poor. The opposite case would similarly deliver unacceptable results.

In the first task besides the classification accuracy, the model parameter such as an activation function and the accordingly estimated probabilities play a significant role for the calculation of weighted revenue. For instance, if we assume the case where the classification was good, but the probabilities are relatively high or low (say, near 50%), the estimation of total revenue calculated by weighted method may be wrong.

The choice of the best models was made based on three criteria: the highest AUC value, the lowest MAE for weighted quarterly revenue and the lowest MAE for unweighted quarterly revenue. Another metrics like accuracy and MAE for closing dates predictions were also compared to each other, but not considered for the choice of best model for the reasons specified later in section 6.2.

Equivalently to ML predictions, the values guessed by sales managers were also evaluated by nearly all metrics (except unweighted revenue calculation MAE) and compared to real outcome. The evaluation was run through test data for each parameter combination, the best models were picked automatically. For each model its ROC and monthly and quarterly mean errors for all upload dates were plotted to enable some visual analysis.

6.2 Evaluation results

Generally the presented 3-Step strategy outperformed the established approach and the guesses of sales managers by all metrics. NN models had shown the best and the worst performances measured by AUC value for both generated and real data. The models with the worst AUC were the ones with the best unweighted MAE, though they classified all deals as *Lost*. Dangerous here is the `accuracy` metric, which was the highest for this model for real data, but also relatively high for generated data. This may be explained by rather small fraction of *Won* deals in the data. Of course such model can not be considered as the best one.

LOGREG model was for both real and generated data not the best choice and was somewhere in the middle between guessed values and NN predictions by all metrics. But the advantage of LOGREG is that their results are very stable and do not depend so much on the parameter configurations as NN.

Further a more comprehensive analysis of differences in various ML models performance for generated and real data will be analyzed. Generally both LOGREG and NN models with best metrics for both generated and real data had very similar configurations.

The summaries of all metrics are given in the tables 6 and 8. The percentage improvement of the most important metrics are summarized in tables 7 and 9. The unweighted quarterly MAE was not considered for this comparison for several reasons. Firstly, unweighted pipeline computation is not common in revenue forecast and hence no appropriate metric for guessed data was computed. Secondly, the models which were best according to this metric, were basically unusable, and for another models the results of weighted predictions were most satisfying.

6.2.1 Evaluation on generated data

Model	AUC	Precision (0/1)	Recall (0/1)	F1-score (0/1)	Accuracy	MAE closing dates	Quarterly MAE weighted	Quarterly MAE unweighted
Neural Networks								
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (2, 2, 2)	0.706	0.81/0.53	0.88/0.38	0.84/0.45	0.76	38.93	47955.27	42767.10
Solver: LBFGS, Activation Function: RELU, Nodes: (20, 16, 10, 4)	0.696	0.79/0.45	0.87/0.30	0.83/0.36	0.73	38.93	42748.83	43007.36
Solver: SGD, Activation Function: IDENTITY, Nodes: (100, 100)	0.622	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	64135.76	41963.42
Logistic Regression								
Solver: LIBLINEAR, C: 1.0	0.682	0.81/0.48	0.84/0.45	0.82/0.46	0.74	38.92	61681.71	48989.18
Solver: NEWTON-CG, C: 1.0	0.682	0.82/0.49	0.84/0.45	0.83/0.47	0.74	38.92	61664.57	48941.56
Solver: NEWTON-CG, C: 0.9	0.681	0.82/0.48	0.83/0.45	0.82/0.47	0.74	38.92	62150.51	48927.49
Guessed	0.623	0.79/0.39	0.80/0.37	0.79/0.38	0.69	63.33	77206.94	-

Table 6: Best models performances for generated data

The best NN and LOGREG models based on their AUC and quarterly MAE measures are compared in the table 6. The summary of percentage improvements in comparison to guessed values are given in the table 7. One may have observed that MAE for closing days predictions is almost identical for all models. The performances of LOGREG models do not differ much and are in general worse that of NN models, but still better than guessed values.

The MAE for closing dates of 63.33 for closing data means, that guesses of sales manager were in average about two months away from the real closing date. ML models have still an error of more than one month, but the improvement measures about 38.5% in comparison with guessed values. After all, the improvement of quarterly MAE is significant (more that 44%), especially considering only slight increase of the AUC value in comparison to guessed probabilities.

Model	**AUC**	**Accuracy**	**MAE closing dates**	**Quarterly MAE weighted**
Neural Networks				
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (2, 2, 2)	13.32%	10.14%	-38.52%	-37.89%
Solver: LBFGS, Activation Function: RELU, Nodes: (20, 16, 10, 4)	11.72%	5.79%	-38.52%	-44.63%
Solver: SGD, Activation Function: IDENTITY, Nodes: (100, 100)	-0.16%	7.24%	-38.54%	-16.93%
Logistic Regression				
Solver: LIBLINEAR, C: 1.0	9.47%	7.24%	-38.54%	-19.85%
Solver: NEWTON-CG, C: 1.0	9.47%	7.24%	-38.54%	-20.13%
Solver: NEWTON-CG, C: 0.9	9.3%	7.24%	-38.54%	-19.5%

Table 7: Best models performance compared to guessed values for generated data

The NN model with the Stochastic Gradient Descent (SGD) solver has the best quarterly unweighted MAE and good MAE for closing dates predictions, but by the closer look one may notice that during classification task all items were ordered to one class and AUC value is even slightly lower than for predictions "guessed" by the imaginary sales manager (don't forget that the data is fake). Therefore we can not consider this model further as best one.

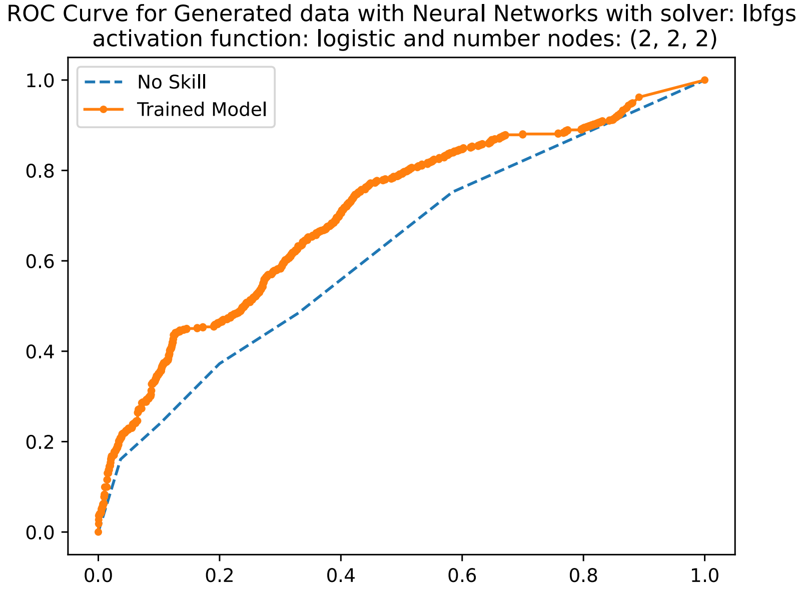


Figure 7: ROC for the model with the highest AUC for generated data

Surprisingly, the model with the best AUC (see figure 7) and also the best accuracy does not have the best quarterly MAE. This could be explained by activation function and difference in probability estimations for classification task. A short look at the unweighted quarterly MAE shows that the difference between unweighted and weighted revenue is stronger for *logistic* activation function (see 4), however for *relu* activation function (5) the values are very close to each other. This fact makes the decision making of the best model to even more challenging task. The comparison of MAE for guessed predictions, weighted pipeline and unweighted pipeline for the model with the lowest MAE is shown in the figure 8. Here one may see how close the outputs of *relu* activation function are to the binary outputs used for the calculation of unweighted pipeline.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

$$f(x) = max(0, x) \tag{5}$$

The problem with logistic activation function may also explain the overall relatively poor performance of the LOGREG models. In the table 6 the best models of that class were picked, but except very tiny improvement in MAE for closing dates predictions in comparison to NN models the results are not that satisfying. The quarterly weighted MAE is only around 20% better than guessed values, and even if we compare the unweighted revenue forecast errors, which are much better, they are still larger than in NN models.

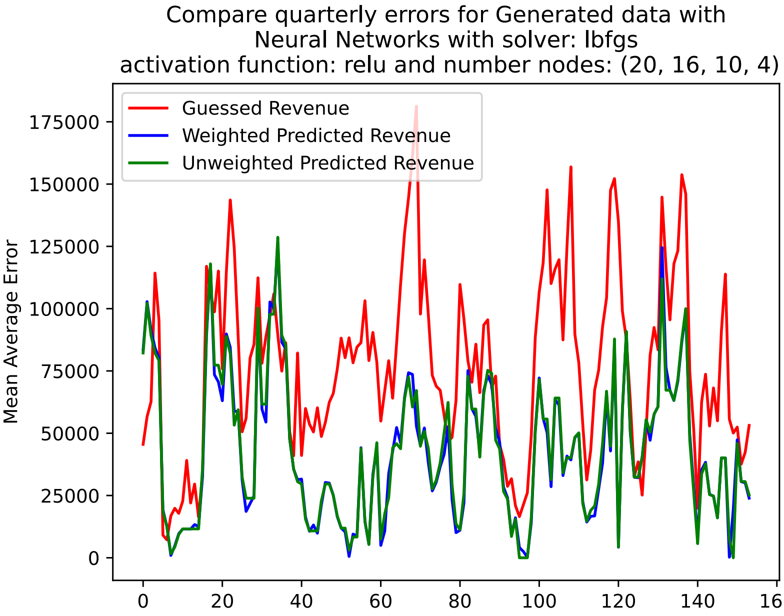


Figure 8: Quarterly revenue MAE for the model with the lowest quarterly revenue MAE for generated data

6.2.2 Evaluation on real data

The accuracy metrics for real data were very similar to generated data with the exception that the errors were much larger. Probably the reason is, that the generated data represents a perfect sales manager who always handles her data the same way and updates the stages regularly. The reality looks often much differently. It can not be guaranteed that even the closing dates were entered in the system properly, which may explain extremely large MAE for closing dates estimations. The summary of all metrics and their percentage comparison to the values entered by sales managers are given in the tables 8 and 9 respectively.

Model	AUC	Precision (0/1)	Recall (0/1)	F1-score (0/1)	Accuracy	MAE closing dates	Quarterly MAE weighted	Quarterly MAE unweighted
Neural Networks								
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (2, 2, 2)	0.786	0.93/0.35	0.88/0.50	0.91/0.41	0.84	133.41	412246.34	381795.60
Solver: LBFGS, Activation Function: RELU, Nodes: (20, 16, 10, 4)	0.769	0.93/0.34	0.89/0.46	.91/0.39	0.84	127.04	241465.41	244019.52
Solver: SGD, Activation Function: IDENTITY, Nodes: (2, 2, 2)	0.668	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	937699.56	168683.89
Logistic Regression								
Solver: LIBLINEAR, C: 1.0	0.751	0.92/0.25	0.83/0.46	0.87/0.32	0.79	134,17	520491.35	470650.59
Guessed	0.704	0.95/0.19	0.57/0.78	0.72/0.30	0.60	185.20	1149879.72	-

Table 8: Best models performances for real data

Similarly to generated data, one model with all-zero predictions presented itself as the best in terms of unweighted quarterly revenue MAE and accuracy. Again a case where a look at AUC value is worth before choosing the best model for further usage. Equivalently to the case with generated data, it was a model with SGD solver. For LOGREG models here only one model of this class was the best in all three relevant metrics. But again, the LOGREG model is still not that good as NN.

Model	AUC	Accuracy	MAE closing dates	Quarterly MAE weighted
Neural Networks				
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (2, 2, 2)	11,65%	40%	-28.19%	-64.15%
Solver: LBFGS, Activation Function: RELU, Nodes: (20, 16, 10, 4)	9.23%	40%	-31.4%	-79%
Solver: SGD, Activation Function: IDENTITY, Nodes: (2, 2, 2)	-5.11%	48.33%	-31.04%	-18.45%
Logistic Regression				
Solver: LIBLINEAR, C: 1.0	6.68%	31.67%	-27.55%	-54.74%

Table 9: Best models performance compared to guessed values for real data

As briefly noticed above, the errors in closing date predictions are huge, but ML methods perform about 30% better than subjective guesses of sales managers, who missed the actual closing date in average of about half a year! The AI was in average still about four month next to the actual date, which is of course disappointing value, especially when the turnover needs to be calculated quarterly. These gigantic errors can explain the huge misses of quarterly revenue predictions: the sales managers had an average error in predictions of more than one million euro, which is of course painful for CEO who needs to prepare the company budget. The performance of NN models were much better in this matter: again, the model with rectified linear unit activation function was nearly 80% more accurate than the estimations of sales managers. The plot with comparison of average errors for each upload date are shown in figure 10.

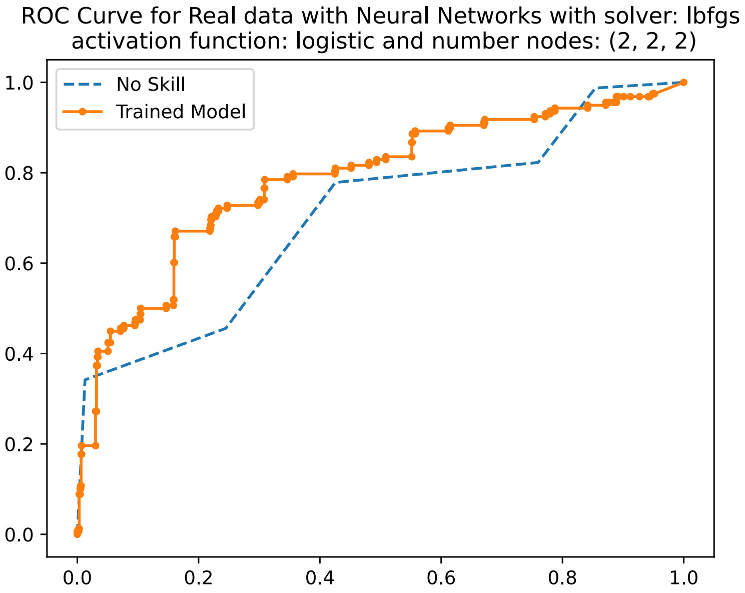


Figure 9: ROC for the model with the highest AUC for real data

The difference between the quarterly MAE weighted revenues for the NN models with logistic or rectified linear unit activation function is much larger for real data. Also MAE for closing dates is slightly different between the models, in contrary to the predictions for generated data. In this case the model with rectified linear unit activation function is clearly better than the one with logistic activation function, even though the latter has slightly better AUC value.

In the figure 10 the ROC for the winning probabilities set by sales manager and the predicted classification probabilities are compared. In some points the curve of sales managers is even higher than the model. This could be interpreted the way that basically the sales managers are not that bad in their guesses of the outcome of the sales opportunity and the winning probability. But the huge errors in closing dates predictions unfortunately have a bad influence on the revenue predictions, which makes the budget planning to a tough task.

6.3 Discussion

Generally ML models performed better than guessed values for both generated and real data. Though the error for quarterly revenue is still quite large in all cases, the ML models confirmed a significant improvement. However the results of NN models were better than for LOGREG, the latter may be preferred due to it's stability. As shown in appendixes A and B, NN models react much more sensitively to parameter settings and often give results which make no sense. Hence a

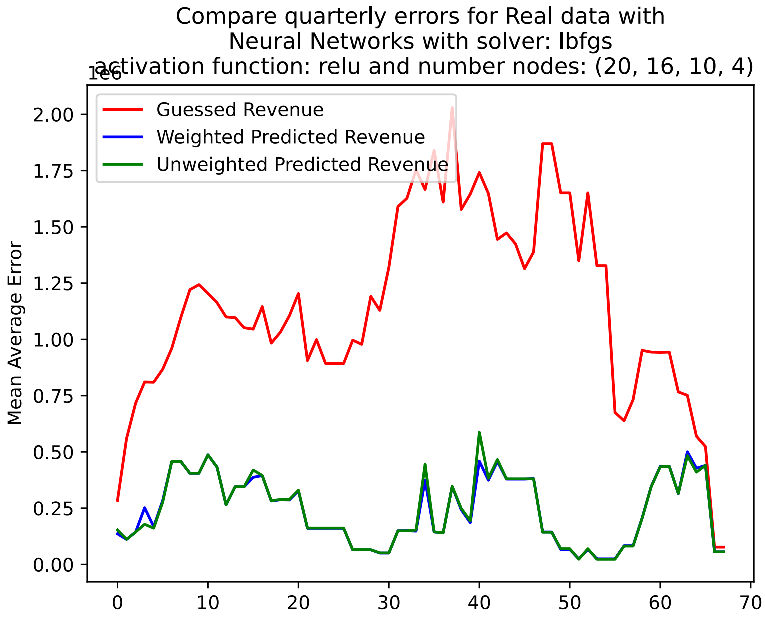


Figure 10: Quarterly revenue MAE for the model with the lowest MAE for real data

cross-validation with lots of different parameter settings is advisable, which would expend training time significantly. On the other side, much better results may be achieved while using NN models for making the forecasts.

Though in this case the best NN models configurations were identical, it may happen that for different data another parameter settings would perform best. To my surprise, the small NN performed better. Probably another configurations for number of layers and nodes could perform even better.

Another insight was that weighted pipeline calculation is preferable even for ML models. It helps to filter out nonsense models, but for somewhat good models with `relu` activation function (5) the results are still very close to unweighted calculations as shown in figures 8 and 10.

7 Conclusions and Future Work

7.1 Summary

Though the AI presented in this thesis is still not very intelligent and generally the machines can not replace humans yet, some routine tasks could be managed by AI. This paper as well as another studies discussed in section 3 confirmed that algorithms can classify the Opportunities much better than sales managers. As interview with the executive of *SmartPM Solutions* Sebastian Wallner has shown that the main problem here is that sales managers often do not have time of willingness to estimate the probabilities for each particular Opportunity manually and therefore set values based only on sales stage or shift the closing dates arbitrarily. The algorithms however are not busy with other tasks and consider all features and therefore can estimate the probabilities more accurately.

The largest problem the algorithms had was the estimation of the closing dates, which may be explained by poorly handled data. Sometimes sales managers do not mark Opportunities as closed properly or forget to update the stage. This is the bad news for sales managers: this work still has to be done appropriately to ensure that the algorithm is fitted with correct data.

7.2 Limitations

Since the algorithm was tested on my local machine, the number of various models configurations was limited.

The generated data was not quite representative, still most features were normally distributed and only few of them had somewhat impact on the outcome.

The models are very sensitive to the quality of input data, due to the fact that preprocessing step was minimized.

Only one real data set was on my disposal, meaning that the results for data from another company could be very different. In addition, the real data was relatively small, containing only 16 months of uploads.

The relatively small size of data allowed to make predictions using `Sklearn` library and perform OHE without having any memory issues. The larger size of data would inevitably lead to memory errors and therefore to necessity of dimensionality reduction and/or usage of parallel algorithm implementations.

7.3 Future work

To improve the quality of predictions in future I would insert a more extensive feature engineering step. The data about clients can be found in the web or from database in case when the data was collected previously, and be further used for making predictions.

More exhaustive preprocessing could help to improve the results for closing date predictions. Here one could think not only about deleting some nonsense values, but also how to automatically detect and eliminate outliers.

Also, outlier detection could be tested as a strategy to find Opportunities which will be won for data with very small percentage of successfull sells.

References

[1] 1.11. ensemble methods. `https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees`. Accessed: 2020-05-30.

[2] 1.17. neural network models (supervised). `https://scikit-learn.org/stable/modules/neural_networks_supervised.html`. Accessed: 2020-05-30.

[3] 3.2.4.3.2. sklearn.ensemble.randomforestregressor. `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html`. Accessed: 2020-06-11.

[4] sklearn.linear_model.logisticregression. `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn.linear_model.LogisticRegression`. Accessed: 2020-05-30.

[5] sklearn.neural_network.mlpclassifier. `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html`. Accessed: 2020-05-30.

[6] Bahari, T. F., and Elayidom, M. S. An efficient crm-data mining framework for the prediction of customer behaviour. *Procedia Computer Science 46*, C (2015), 725–731.

[7] Bauer, E. The weighted sales pipeline: How it works and how to create one for your company. `https://www.propellercrm.com/blog/weighted-sales-pipeline`, 2019. Accessed 13-January-2020.

[8] Bohanec, M., Kljajić Borštnar, M., and Robnik-Šikonja, M. Explaining machine learning models in sales predictions. *Expert Systems With Applications 71* (2017), 416–428.

[9] Bohanec, M., Robnik-Šikonja, M., and Borštnar, M. Organizational learning supported by machine learning models coupled with general explanation methods: A case of b2b sales forecasting. *Organizacija 50*, 3 (2017), 217–233.

[10] Brynjolfsson, E., Hitt, L. M., and Kim, H. H. Strength in numbers: How does data-driven decisionmaking affect firm performance? *Available at SSRN 1819486* (2011).

[11] Cefkin, M. Numbers may speak louder than words, but is anyone listening? the rhythmscape and sales pipeline management. *Ethnographic Praxis in Industry Conference Proceedings 2007*, 1 (2007), 187–199.

[12] D'haen, J., and Van Den Poel, D. Model-supported business-to-business prospect prediction based on an iterative customer acquisition framework. *Industrial Marketing Management 42*, 4 (2013), 544–551.

[13] Eitle, V., and Buxmann, P. Business analytics for sales pipeline management in the software industry: A machine learning perspective. *Proceedings of the 52nd Hawaii International Conference on System Sciences* (2019), 1013–1022.

[14] Gentsch, P. *Künstliche Intelligenz für Sales, Marketing und Service: Mit AI und Bots zu einem Algorithmic Business – Konzepte und Best Practices*, 2. aufl. 2019 ed. Springer Fachmedien Wiesbaden, Wiesbaden, 2019.

[15] Gong, M., Sun, C., Huang, J., and Chu, S. Sales pipeline win propensity prediction: a regression approach. *arXiv.org* (2015).

[16] Holland, K. Sales pipeline stages: A breakdown. `https://www.copper.com/blog/pipeline-stages`, November 2019. Accessed: 2020-06-11.

[17] Lilien, G. L. The b2b knowledge gap. *International Journal of Research in Marketing 33*, 3 (2016), 543–556.

[18] Luxhøj, J. T., Riis, J. O., and Stensballe, B. A hybrid econometric—neural network modeling approach for sales forecasting. *International Journal of Production Economics 43*, 2 (1996), 175 – 192.

[19] Monat, J. P. Industrial sales lead conversion modeling. *Marketing Intelligence & Planning 29*, 2 (2011), 178–194.

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

[21] Salesmate. 7 sales pipeline stages in crm that make your small business a success. `https://medium.com/@SalesmateIO/7-sales-pipeline-stages-in-crm-for-your-successful-small-business-e8bd25d7498c`, October 2018. Accessed: 2020-06-11.

[22] Saunders, A. The 7 sales pipeline stages that every small business should use. `https://keap.com/business-success-blog/marketing/automation/the-7-sales-pipeline-stages-every-small-business-should-use`, March 2020. Accessed: 2020-06-11.

[23] Van Rossum, G., and Drake, F. L. *Python 3 Reference Manual.* CreateSpace, Scotts Valley, CA, 2009.

[24] Yan, J., Gong, M., Sun, C., Huang, J., and Chu, S. M. Sales pipeline win propensity prediction: a regression approach. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)* (2015), IEEE, pp. 854–857.

[25] Yan, J., Zhang, C., Zha, H., Gong, M., Changhua, S., Huang, J., Chu, S., and Yang, X. On machine learning towards predictive sales pipeline analysis. *Twenty-Ninth AAAI Conference on Artificial Intelligence* (2015), 1945–1951.

Glossary

Deal Same as Opportunity. 3, 7–9, 11, 12

Lead A lead is an individual who's at the top of the funnel and hasn't yet been qualified. For example, they might have downloaded a piece of content like a white paper or an eBook or they were contacted by a sales rep via a cold call. [2]. 7, 8

Opportunity A sales opportunity is a qualified prospect who has a high probability of becoming a customer. An opportunity should have a pain point your product or service can solve and an interest in the offering. Salespeople should ensure the opportunity is a good-fit for what they're selling. [3]. 3–9, 12, 25, 26, 29

Acronyms

AI Artifical Intelligence. 7, 8, 23, 25

AUC Area Under the Curve. 18–21, 23, 24

B2B Business-To-Business. 3, 5–8

B2C Business-To-Consumer. 3, 5

CEO Chief Executive Officer. 3, 5, 6, 10, 11, 23

CRM Customer Relationship Management. 3, 4, 12, 13

DDD Data-Driven Decision making. 3

DT Decision Tree. 7, 8

LINREG Linear Regression. 6, 15

LOGREG Logistic Regression. 3, 6–9, 15–17, 20, 22–24

MAE Mean Absolute Error. 6, 18–25

ML Machine Learning. 3, 5–8, 15, 19, 20, 23–25

NN Neural Networks. 3, 6–9, 15–17, 20–25

OHE One-Hot-Encoding. 14, 26

RF Random Forest Regression. 3, 6–8, 15, 17

[2] `https://blog.hubspot.com/sales/criteria-to-upgrade-a-lead-to-an-opportunity-and-theyre-not-what-you-think`

[3] `https://blog.hubspot.com/sales/criteria-to-upgrade-a-lead-to-an-opportunity-and-theyre-not-what-you-think`

ROC Receiver Operating Characteristic Curve. 18, 19, 21, 24

SGD Stochastic Gradient Descent. 21, 23

SVC Support Vector Classifier. 6, 7, 9

Appendices

A Performance summary for generated data

A.1 Neural Networks

Model	AUC	Precision (0/1)	Recall (0/1)	F1-score (0/1)	Acc	MAE closing dates	Quarterly MAE weighted	Quarterly MAE unweighted
GUESSED	*0.623*	*0.79/0.39*	*0.80/0.37*	*0.79/0.38*	*0.69*	*63.33*	*77206.94*	-
Solver: LBFGS, Activation Function: IDENTITY, Nodes: (2, 2, 2)	0.660	0.79/0.46	0.87/0.32	0.83/0.38	0.73	38.93	43947.42	43701.41
Solver: LBFGS, Activation Function: IDENTITY, Nodes: (100, 100)	0.636	0.80/0.48	0.87/0.35	0.83/0.41	0.74	38.92	45497.12	45681.82
Solver: LBFGS, Activation Function: IDENTITY, Nodes: (20, 16, 10, 4)	0.648	0.79/0.46	0.87/0.31	0.83/0.37	0.73	38.92	44850.54	45218.18
Solver: LBFGS, Activation Function: IDENTITY, Nodes: (100, 80, 60, 40)	0.640	0.80/0.49	0.86/0.38	0.83/0.43	0.72	38.92	46025.13	46203.46
Solver: SGD, Activation Function: IDENTITY, Nodes: (2, 2, 2)	0.682	0.80/0.49	0.87/0.37	0.83/0.42	0.74	38.93	51075.28	47666.34
Solver: SGD, Activation Function: IDENTITY, Nodes: (100, 100)	0.622	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	64135.76	41963.42
Solver: SGD, Activation Function: IDENTITY, Nodes: (20, 16, 10, 4)	0.553	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	67283.11	41963.42
Solver: SGD, Activation Function: IDENTITY, Nodes: (100, 80, 60, 40)	0.644	0.79/0.46	0.86/0.35	0.82/0.39	0.73	38.92	49061.75	45706.06
Solver: ADAM, Activation Function: IDENTITY, Nodes: (2, 2, 2)	0.666	0.80/0.53	0.89/0.37	0.84/0.43	0.75	38.93	52851.39	45544.37
Solver: ADAM, Activation Function: IDENTITY, Nodes: (100, 100)	0.659	0.81/0.51	0.87/0.39	0.84/0.44	0.75	38.93	56234.44	46348.59
Solver: ADAM, Activation Function: IDENTITY, Nodes: (20, 16, 10, 4)	0.665	0.80/0.46	0.84/0.40	0.82/0.43	0.73	38.93	57021.89	51142.10
Solver: ADAM, Activation Function: IDENTITY, Nodes: (100, 80, 60, 40)	0.624	0.79/0.45	0.85/0.36	0.82/0.40	0.72	38.91	52460.69	48780.84
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (2, 2, 2)	0.706	0.81/0.53	0.88/0.38	0.84/0.45	0.76	38.93	47955.27	42767.10
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (100, 100)	0.686	0.80/0.50	0.87/0.38	0.83/0.43	0.74	38.92	43253.04	42897.73
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (20, 16, 10, 4)	0.495	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	65115.57	41963.42
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (100, 80, 60, 40)	0.484	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	65116.88	41963.42
Solver: SGD, Activation Function: LOGISTIC, Nodes: (2, 2, 2)	0.497	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	65116.88	41963.42
Solver: SGD, Activation Function: LOGISTIC, Nodes: (100, 100)	0.477	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	71724.34	41963.42
Solver: SGD, Activation Function: LOGISTIC, Nodes: (20, 16, 10, 4)	0.525	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	86569.26	41963.42

Model	**AUC**	**Precision (0/1)**	**Recall (0/1)**	**F1-score (0/1)**	**Acc**	**MAE closing dates**	**Quarterly MAE weighted**	**Quarterly MAE unweighted**
Solver: SGD, Activation Function: LOGISTIC, Nodes: (100, 80, 60, 40)	0.490	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	76987.59	41963.42
Solver: ADAM, Activation Function: LOGISTIC, Nodes: (2, 2, 2)	0.601	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	87706.63	41963.42
Solver: ADAM, Activation Function: LOGISTIC, Nodes: (100, 100)	0.599	0.80/0.45	0.84/0.38	0.82/0.41	0.72	38.93	60027.46	52408.33
Solver: ADAM, Activation Function: LOGISTIC, Nodes: (20, 16, 10, 4)	0.564	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	85089.57	41963.42
Solver: ADAM, Activation Function: LOGISTIC, Nodes: (100, 80, 60, 40)	0.659	0.80/0.45	0.83/0.40	0.82/0.42	0.72	38.931	53728.25	51424.24
Solver: LBFGS, Activation Function: TANH, Nodes: (2, 2, 2)	0.628	0.80/0.33	0.62/0.54	0.69/0.41	0.60	38.92	74649.78	74677.92
Solver: LBFGS, Activation Function: TANH, Nodes: (100, 100)	0.637	0.79/0.48	0.87/0.34	0.83/0.40	0.74	38.94	45959.10	45801.19
Solver: LBFGS, Activation Function: TANH, Nodes: (20, 16, 10, 4)	0.592	0.79/0.39	0.78/0.41	0.78/0.40	0.68	38.92	53558.07	53546.00
Solver: LBFGS, Activation Function: TANH, Nodes: (100, 80, 60, 40)	0.658	0.79/0.39	0.79/0.39	0.79/0.39	0.69	38.92	55062.39	55405.30
Solver: SGD, Activation Function: TANH, Nodes: (2, 2, 2)	0.538	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	73280.64	41963.42
Solver: SGD, Activation Function: TANH, Nodes: (100, 100)	0.507	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	74866.07	41963.42
Solver: SGD, Activation Function: TANH, Nodes: (20, 16, 10, 4)	0.539	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	70968.31	41963.42
Solver: SGD, Activation Function: TANH, Nodes: (100, 80, 60, 40)	0.651	0.79/0.44	0.85/0.33	0.82/0.38	0.72	38.94	53321.73	44854.76
Solver: ADAM, Activation Function: TANH, Nodes: (2, 2, 2)	0.674	0.80/0.55	0.90/0.36	0.85/0.44	0.76	38.93	63161.19	5016.88
Solver: ADAM, Activation Function: TANH, Nodes: (100, 100)	0.659	0.81/0.52	0.88/0.39	0.84/0.44	0.75	38.93	55894.63	46246.43
Solver: ADAM, Activation Function: TANH, Nodes: (20, 16, 10, 4)	0.666	0.80/0.46	0.84/0.39	0.82/0.42	0.73	38.93	56149.49	50103.68
Solver: ADAM, Activation Function: TANH, Nodes: (100, 80, 60, 40)	0.626	0.80/0.44	0.84/0.38	0.82/0.41	0.72	38.91	53572.80	51146.65
Solver: LBFGS, Activation Function: RELU, Nodes: (2, 2, 2)	0.500	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	65116.85	41963.42
Solver: LBFGS, Activation Function: RELU, Nodes: (100, 100)	0.657	0.80/0.49	0.86/0.38	0.83/0.43	0.74	38.92	45879.66	45432.47
Solver: LBFGS, Activation Function: RELU, Nodes: (20, 16, 10, 4)	0.696	0.78/0.45	0.87/0.30	0.83/0.36	0.73	38.93	42748.83	43007.36
Solver: LBFGS, Activation Function: RELU, Nodes: (100, 80, 60, 40)	0.654	0.80/0.48	0.87/0.37	0.83/0.42	0.74	38.93	46557.76	46103.57
Solver: SGD, Activation Function: RELU, Nodes: (2, 2, 2)	0.500	0.00/0.26	0.00/1.00	0.00/0.41	0.26	38.97	128695.62	164502.16
Solver: SGD, Activation Function: RELU, Nodes: (100, 100)	0.535	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	80725.98	41963.42

Model	**AUC**	**Precision (0/1)**	**Recall (0/1)**	**F1-score (0/1)**	**Acc**	**MAE closing dates**	**Quarterly MAE weighted**	**Quarterly MAE unweighted**
Solver: SGD, Activation Function: RELU, Nodes: (20, 16, 10, 4)	0.519	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	81064.39	41963.42
Solver: SGD, Activation Function: RELU, Nodes: (100, 80, 60, 40)	0.491	0.74/0.00	1.00/0.00	0.85/0.00	0.74	38.92	81341.80	41963.42
Solver: ADAM, Activation Function: RELU, Nodes: (2, 2, 2)	0.500	0.00/0.26	0.00/1.00	0.00/0.41	0.26	38.97	130002.19	164502.16
Solver: ADAM, Activation Function: RELU, Nodes: (100, 100)	0.669	0.80/0.44	0.81/0.42	0.81/0.43	0.71	38.92	60033.44	53075.97
Solver: ADAM, Activation Function: RELU, Nodes: (20, 16, 10, 4)	0.669	0.81/0.46	0.81/0.46	0.81/0.46	0.72	38.92	63265.57	54243.40
Solver: ADAM, Activation Function: RELU, Nodes: (100, 80, 60, 40)	0.642	0.81/0.45	0.81/0.45	0.81/0.45	0.72	38.92	53557.73	52360.71

Table 10: Neural Network models performances for generated data

A.2 Logistic Regression

Model	**AUC**	**Precision (0/1)**	**Recall (0/1)**	**F1-score (0/1)**	**Acc**	**MAE closing dates**	**Quarterly MAE weighted**	**Quarterly MAE unweighted**
GUESSED	*0.704*	*0.95/0.19*	*0.57/0.78*	*0.72/0.30*	*0.60*	*185.20*	*1149879.72*	-
Solver: NEWTON-SG, C: 0.1	0.667	0.83/0.42	0.73/0.56	0.78/0.48	0.69	38.91	74297.03	59825.65
Solver: NEWTON-SG, C: 0.2	0.674	0.82/0.42	0.76/0.51	0.79/0.46	0.69	38.92	70254.38	55849.78
Solver: NEWTON-SG, C: 0.3	0.678	0.82/0.43	0.78/0.49	0.80/0.46	0.71	38.92	67933.23	53369.59
Solver: NEWTON-SG, C: 0.4	0.679	0.82/0.45	0.80/0.48	0.81/0.46	0.72	38.92	66330.82	51960.06
Solver: NEWTON-SG, C: 0.5	0.680	0.82/0.47	0.82/0.47	0.82/0.47	0.73	38.92	65109.70	50758.98
Solver: NEWTON-SG, C: 0.6	0.680	0.82/0.48	0.82/0.46	0.82/0.47	0.73	38.92	64143.68	49929.55
Solver: NEWTON-SG, C: 0.7	0.681	0.82/0.48	0.82/0.46	0.82/0.47	0.73	38.92	63365.67	49274.24
Solver: NEWTON-SG, C: 0.8	0.681	0.82/0.48	0.83/0.45	0.82/0.47	0.74	38.92	62709.10	49079.11
Solver: NEWTON-SG, C: 0.9	0.681	0.82/0.48	0.83/0.45	0.82/0.47	0.74	38.92	62150.51	48927.49
Solver: NEWTON-SG, C: 1.0	0.682	0.82/0.49	0.84/0.45	0.83/0.47	0.74	38.92	61664.57	48941.56
Solver: LBFGS, C: 0.1	0.667	0.83/0.42	0.73/0.56	0.78/0.48	0.69	38.91	74296.25	59797.08
Solver: LBFGS, C: 0.2	0.674	0.82/0.42	0.76/0.51	0.79/0.46	0.69	38.92	70254.00	55849.78
Solver: LBFGS, C: 0.3	0.678	0.82/0.43	0.78/0.49	0.80/0.46	0.71	38.92	67933.53	53369.59
Solver: LBFGS, C: 0.4	0.679	0.82/0.45	0.80/0.48	0.81/0.46	0.72	38.92	66331.78	51953.35
Solver: LBFGS, C: 0.5	0.680	0.82/0.47	0.82/0.47	0.82/0.47	0.73	38.92	65109.19	50758.98
Solver: LBFGS, C: 0.6	0.680	0.82/0.48	0.82/0.46	0.82/0.47	0.73	38.92	64146.58	49929.55
Solver: LBFGS, C: 0.7	0.681	0.82/0.48	0.82/0.46	0.82/0.47	0.73	38.92	63363.37	49274.24
Solver: LBFGS, C: 0.8	0.681	0.82/0.48	0.83/0.45	0.82/0.47	0.74	38.92	62710.32	49079.11
Solver: LBFGS, C: 0.9	0.681	0.82/0.48	0.83/0.45	0.82/0.47	0.74	38.92	62153.29	48927.49
Solver: LBFGS, C: 1.0	0.681	0.82/0.49	0.84/0.45	0.83/0.47	0.74	38.92	61665.87	48941.56
Solver: LIBLINEAR, C: 0.1	0.667	0.83/0.41	0.73/0.56	0.78/0.47	0.69	38.91	74301.48	60057.03
Solver: LIBLINEAR, C: 0.2	0.674	0.82/0.42	0.76/0.51	0.79/0.46	0.69	38.92	70272.70	56048.81
Solver: LIBLINEAR, C: 0.3	0.678	0.82/0.43	0.78/0.49	0.80/0.46	0.71	38.92	67954.65	53632.79
Solver: LIBLINEAR, C: 0.4	0.680	0.82/0.46	0.81/0.47	0.81/0.46	0.72	38.92	66352.22	51403.68

Model	AUC	Precision (0/1)	Recall (0/1)	F1-score (0/1)	Acc	MAE closing dates	Quarterly MAE weighted	Quarterly MAE unweighted
Solver: LIBLINEAR, C: 0.5	0.680	0.82/0.47	0.82/0.47	0.82/0.47	0.73	38.92	65131.38	50431.39
Solver: LIBLINEAR, C: 0.6	0.680	0.82/0.48	0.83/0.46	0.82/0.47	0.73	38.92	64165.50	49371.21
Solver: LIBLINEAR, C: 0.7	0.681	0.82/0.48	0.83/0.46	0.82/0.47	0.73	38.92	63385.00	49180.95
Solver: LIBLINEAR, C: 0.8	0.681	0.82/0.48	0.83/0.45	0.82/0.47	0.73	38.92	62728.14	49253.79
Solver: LIBLINEAR, C: 0.9	0.682	0.82/0.49	0.84/0.45	0.83/0.47	0.74	38.92	62167.91	49173.38
Solver: LIBLINEAR, C: 1.0	0.682	0.81/0.48	0.84/0.45	0.82/0.46	0.74	38.92	61681.71	48989.18
Solver: SAG, C: 0.1	0.667	0.83/0.42	0.73/0.56	0.78/0.48	0.69	38.91	74296.56	59797.08
Solver: SAG, C: 0.2	0.674	0.82/0.42	0.76/0.51	0.79/0.46	0.69	38.92	70253.40	55849.78
Solver: SAG, C: 0.3	0.678	0.82/0.43	0.78/0.49	0.80/0.46	0.71	38.92	67933.45	53369.59
Solver: SAG, C: 0.4	0.679	0.82/0.45	0.80/0.48	0.81/0.46	0.72	38.92	66328.07	51960.06
Solver: SAG, C: 0.5	0.680	0.82/0.47	0.82/0.47	0.82/0.47	0.73	38.92	65111.22	50758.98
Solver: SAG, C: 0.6	0.680	0.82/0.48	0.82/0.46	0.82/0.47	0.73	38.92	64145.92	49929.55
Solver: SAG, C: 0.7	0.681	0.82/0.48	0.83/0.46	0.82/0.47	0.73	38.92	63361.58	49274.24
Solver: SAG, C: 0.8	0.681	0.82/0.48	0.83/0.45	0.82/0.47	0.74	38.92	62704.53	49079.11
Solver: SAG, C: 0.9	0.681	0.82/0.48	0.83/0.45	0.82/0.47	0.74	38.92	62149.41	48927.49
Solver: SAG, C: 1.0	0.681	0.82/0.49	0.84/0.45	0.83/0.47	0.74	38.92	61670.81	48941.56
Solver: SAGA, C: 0.1	0.667	0.83/0.42	0.73/0.56	0.78/0.48	0.69	38.91	74296.07	59825.65
Solver: SAGA, C: 0.2	0.674	0.82/0.42	0.76/0.51	0.79/0.46	0.69	38.92	70254.80	55849.78
Solver: SAGA, C: 0.3	0.678	0.82/0.43	0.78/0.49	0.80/0.46	0.71	38.92	67933.70	53369.59
Solver: SAGA, C: 0.4	0.679	0.82/0.45	0.80/0.48	0.81/0.46	0.72	38.92	66331.14	51878.46
Solver: SAGA, C: 0.5	0.680	0.82/0.47	0.82/0.47	0.82/0.47	0.73	38.92	65110.59	50758.98
Solver: SAGA, C: 0.6	0.680	0.82/0.48	0.82/0.46	0.82/0.47	0.73	38.92	64144.33	49929.55
Solver: SAGA, C: 0.7	0.681	0.82/0.48	0.83/0.46	0.82/0.47	0.73	38.92	63365.80	49274.24
Solver: SAGA, C: 0.8	0.681	0.82/0.48	0.83/0.45	0.82/0.47	0.74	38.92	62710.44	49079.11
Solver: SAGA, C: 0.9	0.681	0.82/0.48	0.83/0.45	0.82/0.47	0.74	38.92	62151.06	48927.49
Solver: SAGA, C: 1.0	0.681	0.82/0.49	0.84/0.45	0.83/0.47	0.74	38.92	61667.32	48941.56

Table 11: Logistic Regression models performances for generated data

B Performance summary for real data

B.1 Neural Networks

Model	AUC	Precision (0/1)	Recall (0/1)	F1-score (0/1)	Acc	MAE closing dates	Quarterly MAE weighted	Quarterly MAE unweighted
GUESSED	*0.704*	*0.95/0.19*	*0.57/0.78*	*0.72/0.30*	*0.60*	*185.20*	*1149879.72*	-
Solver: LBFGS, Activation Function: IDENTITY, Nodes: (2, 2, 2)	0.763	0.93/0.26	0.82/0.51	0.87/0.34	0.78	143.28	478184.04	501147.48
Solver: LBFGS, Activation Function: IDENTITY, Nodes: (100, 100)	0.759	0.93/0.33	0.87/0.50	0.90/0.40	0.83	131.97	361584.31	348980.54
Solver: LBFGS, Activation Function: IDENTITY, Nodes: (20, 16, 10, 4)	0.733	0.92/0.36	0.92/0.37	0.92/0.37	0.86	125.27	272043.27	249740.51
Solver: LBFGS, Activation Function: IDENTITY, Nodes: (100, 80, 60, 40)	0.754	0.92/0.29	0.87/0.42	0.90/0.34	0.82	130.28	318077.24	325282.70

Model	**AUC**	**Precision (0/1)**	**Recall (0/1)**	**F1-score (0/1)**	**Acc**	**MAE closing dates**	**Quarterly MAE weighted**	**Quarterly MAE unweighted**
Solver: SGD, Activation Function: IDENTITY, Nodes: (2, 2, 2)	0.668	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	937699.56	168683.89
Solver: SGD, Activation Function: IDENTITY, Nodes: (100, 100)	0.576	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	934700.78	168683.89
Solver: SGD, Activation Function: IDENTITY, Nodes: (20, 16, 10, 4)	0.530	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	832269.86	168683.89
Solver: SGD, Activation Function: IDENTITY, Nodes: (100, 80, 60, 40)	0.426	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	960176.92	168683.89
Solver: ADAM, Activation Function: IDENTITY, Nodes: (2, 2, 2)	0.599	0.89/0.75	1.00/0.06	0.94/0.11	0.89	126.77	1052599.67	169039.03
Solver: ADAM, Activation Function: IDENTITY, Nodes: (100, 100)	0.576	0.91/0.26	0.89/0.32	0.90/0.29	0.82	131.93	427034.82	353934.74
Solver: ADAM, Activation Function: IDENTITY, Nodes: (20, 16, 10, 4)	0.704	0.91/0.22	0.85/0.34	0.88/0.27	0.79	135.09	484146.42	440557.06
Solver: ADAM, Activation Function: IDENTITY, Nodes: (100, 80, 60, 40)	0.739	0.92/0.22	0.82/0.42	0.86/0.29	0.77	137.04	502511.43	494275.06
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (2, 2, 2)	0.786	0.93/0.35	0.88/0.50	0.91/0.41	0.84	133.41	412246.34	381795.60
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (100, 100)	0.718	0.92/0.28	0.87/0.42	0.89/0.34	0.82	133.07	402077.49	381842.80
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (20, 16, 10, 4)	0.462	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	483537.75	168683.89
Solver: LBFGS, Activation Function: LOGISTIC, Nodes: (100, 80, 60, 40)	0.467	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	483771.60	168683.89
Solver: SGD, Activation Function: LOGISTIC, Nodes: (2, 2, 2)	0.643	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	1330165.78	168683.89
Solver: SGD, Activation Function: LOGISTIC, Nodes: (100, 100)	0.565	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	983538.28	168683.89
Solver: SGD, Activation Function: LOGISTIC, Nodes: (20, 16, 10, 4)	0.509	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	1170943.03	168683.89
Solver: SGD, Activation Function: LOGISTIC, Nodes: (100, 80, 60, 40)	0.439	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	1113085.96	168683.89
Solver: ADAM, Activation Function: LOGISTIC, Nodes: (2, 2, 2)	0.601	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	1335668.62	168683.89
Solver: ADAM, Activation Function: LOGISTIC, Nodes: (100, 100)	0.599	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	560063.44	168683.89
Solver: ADAM, Activation Function: LOGISTIC, Nodes: (20, 16, 10, 4)	0.611	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	1151848.99	168683.89
Solver: ADAM, Activation Function: LOGISTIC, Nodes: (100, 80, 60, 40)	0.548	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	1079940.73	168683.89
Solver: LBFGS, Activation Function: TANH, Nodes: (2, 2, 2)	0.770	0.94/0.38	0.89/0.53	0.91/0.44	0.85	126.57	257232.06	254177.15
Solver: LBFGS, Activation Function: TANH, Nodes: (100, 100)	0.758	0.93/0.34	0.90/0.44	0.91/0.38	0.84	129.03	405117.73	300507.74
Solver: LBFGS, Activation Function: TANH, Nodes: (20, 16, 10, 4)	0.748	0.93/0.36	0.90/0.43	0.92/0.39	0.85	127.72	286889.76	296909.30

Model	**AUC**	**Precision (0/1)**	**Recall (0/1)**	**F1-score (0/1)**	**Acc**	**MAE closing dates**	**Quarterly MAE weighted**	**Quarterly MAE unweighted**
Solver: LBFGS, Activation Function: TANH, Nodes: (100, 80, 60, 40)	0.745	0.93/0.31	0.86/0.50	0.89/0.38	0.82	132.27	313586.93	346647.10
Solver: SGD, Activation Function: TANH, Nodes: (2, 2, 2)	0.657	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	1247424.56	168683.89
Solver: SGD, Activation Function: TANH, Nodes: (100, 100)	0.580	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	976471.29	168683.89
Solver: SGD, Activation Function: TANH, Nodes: (20, 16, 10, 4)	0.551	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	897230.57	168683.89
Solver: SGD, Activation Function: TANH, Nodes: (100, 80, 60, 40)	0.428	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	1035439.47	168683.89
Solver: ADAM, Activation Function: TANH, Nodes: (2, 2, 2)	0.588	0.90/0.41	0.97/0.16	0.94/0.23	0.88	123.01	1245394.42	181356.02
Solver: ADAM, Activation Function: TANH, Nodes: (100, 100)	0.727	0.91/0.30	0.91/0.32	0.91/0.31	0.84	131.81	417883.68	309720.12
Solver: ADAM, Activation Function: TANH, Nodes: (20, 16, 10, 4)	0.701	0.91/0.21	0.83/0.36	0.87/0.27	0.78	135.67	609761.77	455673.21
Solver: ADAM, Activation Function: TANH, Nodes: (100, 80, 60, 40)	0.738	0.92/0.22	0.82/0.42	0.87/0.29	0.77	137.22	493986.43	489853.62
Solver: LBFGS, Activation Function: RELU, Nodes: (2, 2, 2)	0.500	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	483630.41	168683.89
Solver: LBFGS, Activation Function: RELU, Nodes: (100, 100)	0.750	0.93/0.35	0.89/0.47	0.91/0.41	0.85	127.69	331902.19	300571.48
Solver: LBFGS, Activation Function: RELU, Nodes: (20, 16, 10, 4)	0.769	0.93/0.34	0.89/0.46	0.91/0.39	0.84	127.04	241465.41	244019.52
Solver: LBFGS, Activation Function: RELU, Nodes: (100, 80, 60, 40)	0.745	0.92/0.34	0.92/0.34	0.92/0.34	0.85	124.99	282194.23	248666.85
Solver: SGD, Activation Function: RELU, Nodes: (2, 2, 2)	0.500	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	969002.52	168683.89
Solver: SGD, Activation Function: RELU, Nodes: (100, 100)	0.667	0.89/0.00	1.00/0.01	0.94/0.01	0.89	127.64	1192996.63	169496.39
Solver: SGD, Activation Function: RELU, Nodes: (20, 16, 10, 4)	0.471	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	1045756.91	168683.89
Solver: SGD, Activation Function: RELU, Nodes: (100, 80, 60, 40)	0.503	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	1192708.91	168683.89
Solver: ADAM, Activation Function: RELU, Nodes: (2, 2, 2)	0.500	0.89/0.00	1.00/0.00	0.94/0.00	0.89	127.71	971857.38	168683.89
Solver: ADAM, Activation Function: RELU, Nodes: (100, 100)	0.741	0.92/0.23	0.84/0.39	0.87/0.29	0.79	133.33	499106.57	475374.39
Solver: ADAM, Activation Function: RELU, Nodes: (20, 16, 10, 4)	0.707	0.90/0.76	0.99/0.16	0.85/0.27	0.90	125.45	318352.00	177969.12
Solver: ADAM, Activation Function: RELU, Nodes: (100, 80, 60, 40)	0.708	0.92/0.25	0.85/0.39	0.89/0.31	0.80	134.57	422066.07	433297.51

Table 12: Neural Network models performances for real data

B.2 Logistic Regression

Model	**AUC**	**Precision (0/1)**	**Recall (0/1)**	**F1-score (0/1)**	**Acc**	**MAE closing dates**	**Quarterly MAE weighted**	**Quarterly MAE unweighted**
GUESSED	*0.704*	*0.95/0.19*	*0.57/0.78*	*0.72/0.30*	*0.60*	*185.20*	*1149879.72*	-
Solver: NEWTON-SG, C: 0.1	0.737	0.93/0.24	0.81/0.49	0.86/0.32	0.77	136.38	671422.20	516549.45
Solver: NEWTON-SG, C: 0.2	0.739	0.92/0.23	0.81/0.45	0.86/0.30	0.77	135.33	612425.90	505190.24
Solver: NEWTON-SG, C: 0.3	0.738	0.92/0.23	0.81/0.46	0.86/0.31	0.77	135.39	582937.21	507031.42
Solver: NEWTON-SG, C: 0.4	0.739	0.92/0.23	0.81/0.46	0.86/0.31	0.77	135.12	564447.19	504360.18
Solver: NEWTON-SG, C: 0.5	0.744	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.50	552192.39	491604.30
Solver: NEWTON-SG, C: 0.6	0.746	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	543091.53	489423.25
Solver: NEWTON-SG, C: 0.7	0.748	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	535981.66	489423.25
Solver: NEWTON-SG, C: 0.8	0.748	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	530218.87	489423.25
Solver: NEWTON-SG, C: 0.9	0.749	0.92/0.24	0.82/0.46	0.87/0.32	0.78	134.18	525429.00	474212.59
Solver: NEWTON-SG, C: 1.0	0.749	0.92/0.24	0.82/0.46	0.87/0.32	0.78	134.00	521362.54	473395.59
Solver: LBFGS, C: 0.1	0.737	0.93/0.24	0.81/0.49	0.86/0.32	0.77	136.38	671417.71	516549.45
Solver: LBFGS, C: 0.2	0.739	0.92/0.23	0.81/0.45	0.86/0.30	0.77	135.33	612423.44	505190.24
Solver: LBFGS, C: 0.3	0.738	0.92/0.23	0.81/0.46	0.86/0.31	0.77	135.39	582937.13	507031.42
Solver: LBFGS, C: 0.4	0.739	0.92/0.23	0.81/0.46	0.86/0.31	0.77	135.12	564441.99	504360.18
Solver: LBFGS, C: 0.5	0.744	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.50	552187.35	491604.30
Solver: LBFGS, C: 0.6	0.746	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	543089.98	489423.25
Solver: LBFGS, C: 0.7	0.748	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	535970.95	489423.25
Solver: LBFGS, C: 0.8	0.748	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	530216.21	489423.25
Solver: LBFGS, C: 0.9	0.749	0.92/0.24	0.82/0.46	0.87/0.32	0.78	134.18	525430.03	474212.59
Solver: LBFGS, C: 1.0	0.749	0.92/0.24	0.82/0.46	0.87/0.32	0.78	134.00	521361.95	473395.59
Solver: LIBLINEAR, C: 0.1	0.738	0.93/0.24	0.81/0.49	0.86/0.32	0.77	136.33	671219.19	516048.16
Solver: LIBLINEAR, C: 0.2	0.742	0.92/0.23	0.81/0.46	0.86/0.31	0.77	135.35	612185.95	506530.13
Solver: LIBLINEAR, C: 0.3	0.742	0.92/0.23	0.81/0.46	0.86/0.31	0.77	135.35	582592.43	506530.13
Solver: LIBLINEAR, C: 0.4	0.743	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.50	563986.84	491604.30
Solver: LIBLINEAR, C: 0.5	0.744	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.50	551578.78	491604.30
Solver: LIBLINEAR, C: 0.6	0.748	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.37	542419.00	489053.76
Solver: LIBLINEAR, C: 0.7	0.749	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.37	535250.66	489053.76
Solver: LIBLINEAR, C: 0.8	0.750	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.36	529438.49	473810.24
Solver: LIBLINEAR, C: 0.9	0.750	0.92/0.24	0.82/0.46	0.87/0.32	0.78	134.36	524615.30	473810.24
Solver: LIBLINEAR, C: 1.0	0.751	0.92/0.25	0.83/0.46	0.87/0.32	0.79	134.17	520491.35	470650.59
Solver: SAG, C: 0.1	0.737	0.93/0.24	0.81/0.49	0.86/0.32	0.77	136.38	671409.15	516549.45
Solver: SAG, C: 0.2	0.739	0.92/0.23	0.81/0.45	0.86/0.30	0.77	135.33	612391.14	505190.24
Solver: SAG, C: 0.3	0.738	0.92/0.23	0.81/0.46	0.86/0.31	0.77	135.39	582934.02	507031.42
Solver: SAG, C: 0.4	0.739	0.92/0.23	0.81/0.46	0.86/0.31	0.77	135.12	564475.77	504360.18
Solver: SAG, C: 0.5	0.744	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.50	552183.75	491604.30
Solver: SAG, C: 0.6	0.746	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	543080.42	489423.25
Solver: SAG, C: 0.7	0.748	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	535967.04	489423.25
Solver: SAG, C: 0.8	0.748	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	530181.83	489423.25
Solver: SAG, C: 0.9	0.749	0.92/0.24	0.82/0.46	0.87/0.32	0.78	134.18	525418.44	474212.59
Solver: SAG, C: 1.0	0.749	0.92/0.24	0.82/0.46	0.87/0.32	0.78	134.00	521358.06	473395.59
Solver: SAGA, C: 0.1	0.737	0.93/0.24	0.81/0.49	0.86/0.32	0.77	136.38	671430.48	516549.45
Solver: SAGA, C: 0.2	0.739	0.92/0.23	0.81/0.45	0.86/0.30	0.77	135.33	612425.17	505190.24
Solver: SAGA, C: 0.3	0.738	0.92/0.23	0.81/0.46	0.86/0.31	0.77	135.39	582935.38	507031.42
Solver: SAGA, C: 0.4	0.739	0.92/0.23	0.81/0.46	0.86/0.31	0.77	135.12	564439.06	504360.18

Model	**AUC**	**Precision (0/1)**	**Recall (0/1)**	**F1-score (0/1)**	**Acc**	**MAE closing dates**	**Quarterly MAE weighted**	**Quarterly MAE unweighted**
Solver: SAGA, C: 0.5	0.744	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.50	552181.11	491604.30
Solver: SAGA, C: 0.6	0.746	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	543082.16	489423.25
Solver: SAGA, C: 0.7	0.748	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	535966.85	489423.25
Solver: SAGA, C: 0.8	0.748	0.92/0.24	0.82/0.46	0.87/0.31	0.78	134.43	530199.52	489423.25
Solver: SAGA, C: 0.9	0.749	0.92/0.24	0.82/0.46	0.87/0.32	0.78	134.18	525402.03	474212.59
Solver: SAGA, C: 1.0	0.749	0.92/0.24	0.82/0.46	0.87/0.32	0.78	134.00	521343.27	473395.59

Table 13: Logistic Regression models performances for real data