Network for Studies on Pensions, Aging and Retirement

NETSPAR ACADEMIC SERIES

Predicting Employee Attrition

A Machine Learning Comparison

Danique Heuten

MSc 05/2021-009

Predicting Employee Attrition: A Machine Learning Comparison

Danique Heuten

SNR: 1279202

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee

Supervisor: prof. dr. E.O. Postma
Second reader: dr. D. Stowell

Tilburg University
School of Humanities & Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
May 2021

Preface

Dear readers,

This thesis on “Predicting Employee Attrition: A Machine Learning Comparison” is the final project to graduate from the Master Data Science and Society at Tilburg University, with a specialization in Business.

I was involved in this project from February 2021 to May 2021.

I would like to thank my supervisor, prof. dr. E.O. Postma, for his guidance, support, and feedback during this period. Furthermore, I would like to thank my family and friends for supporting me. In particular, I would like to thank L. Benneker, M. Hermans, B. Heuten, L. van de Kamp, S. Lindenschot and B. van der Velde.

I hope you enjoy reading my thesis about employee attrition prediction.

Abstract

Employees are essential resources for a company. However, several issues can arise when employees decide to leave the company, such as additional costs and reputation loss. Companies want to avoid these issues. Therefore, there is a growing interest in using machine learning to predict which employee is likely to leave. Whereas existing literature focused on traditional machine learning methods, this thesis introduces a new method for predicting employee attrition, namely neural networks. Therefore, this thesis aims to examine whether and to what extent neural networks outperform traditional machine learning methods in predicting employee attrition. The traditional machine learning methods used in this thesis are K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest (RF). This thesis uses a dataset published by IBM Watson Analytics, which contains 1470 instances and 35 features. With that dataset two experiments are carried out. First, the performance of artificial neural networks (ANNs) is compared to the performance of the traditional machine learning methods on the original imbalanced dataset. Second, a balanced dataset is created by using Synthetic Minority Oversampling Technique (SMOTE). Hereafter, all four methods are trained and compared on the balanced dataset. The first experiment showed that the ANNs outperformed the KNN and RF. However, the ANN did not outperform the SVM. The second experiment showed that the ANN did not outperform any of the traditional methods. Therefore, in this research we can conclude that ANNs do not outperform traditional machine learning methods in predicting employee attrition.

Keywords: *Employee attrition, Machine Learning, KNN, SVM, RF, Artificial Neural Networks (ANNs), Imbalanced classes.*

Table of Contents

1. Introduction

Employees are the key resources of a company, as they are the reasons behind a company's success (Das & Baruah, 2013). When an employee decides to leave the company, there may be severe drawbacks for the organization. First of all, high employee attrition harms the reputation of the company, longstanding strategies of the firm, customer satisfaction, and job satisfaction of remaining employees (Das & Baruah, 2013; Ajit & Punnoose, 2016; Alduayj & Rajpoot, 2018; Khera & Divya, 2019). Moreover, additional costs arise, which can lead to financial losses. For example, costs for finding and hiring a new employee, training costs, and costs that arise from reduced productivity of remaining employees (Alduayj & Rajpoot, 2018; Frye, Boomhower, Smith, Vitovsky, & Fabricant, 2018; Khera & Divya, 2019). Machine learning methods can be used to identify which employee is likely to leave. With that information, the management might be able to prevent the employee from leaving. Therefore, machine learning may help solve employee attrition (Ajit & Punnoose, 2016; Yiğit & Shourabizadeh, 2017).

Despite prior research for predicting employee attrition, there is no consistent answer to what machine learning method best predicts employee attrition (Ajit & Punnoose, 2016; Yiğit & Shourabizadeh, 2017; Alduayj & Rajpoot, 2018; Jain & Nayyar, 2018). Furthermore, unlike in customer churn prediction, artificial neural networks (ANNs) have not been investigated in predicting employee attrition (Tsai & Lu, 2009; Keramati, Jafai-Marandi, Aliannejadi, Ahmadian, Mozaffari, & Abbasi, 2014). Existing research in predicting employee attrition relied on traditional machine learning methods (Ajit & Punnoose, 2016; Yiğit & Shourabizadeh, 2017; Alduayj & Rajpoot, 2018; Jain & Nayyar, 2018). Building on previous research, this thesis will study neural networks as a method for predicting employee attrition. Therefore, the goal of this thesis is to examine whether and to what extent neural networks can outperform traditional machine learning methods in predicting employee attrition, which leads to the following main research question:

RQ1: To what extent do neural networks outperform traditional machine learning methods in predicting employee attrition?

Employee attrition is a rare event, which leads to imbalanced classes. Imbalanced classes can result in inaccurate predictions for the minority class because the method will ignore the minority class and focus on the majority class (Guo, Yin, Dong, Yang, & Zhou, 2008; Zhu, Baesens, & van den Broucke, 2017; Patel et al., 2020). Therefore, in this thesis we will conduct two experiments to take the effect of imbalanced classes into account. First, in this thesis we will examine whether and to what extent neural networks outperform traditional machine learning methods in predicting employee attrition, by using the original imbalanced dataset. Second, in this thesis we will examine the performance of neural networks compared to traditional machine learning methods in predicting employee attrition on a balanced dataset by using Synthetic Minority Oversampling Technique (SMOTE). As a result, this thesis splits up the main research question into two sub-questions:

RQ1.a: To what extent do neural networks outperform traditional machine learning methods in predicting employee attrition when the dataset is imbalanced?

R1Q.b: To what extent do neural networks outperform traditional machine learning methods in predicting employee attrition when the dataset is balanced?

To provide answers to these research questions, two experiments are carried out. First, four methods are trained on the imbalanced dataset. The four methods are three traditional machine learning methods (KNN, SVM, RF) and an ANN. Second, to take the class imbalance into account, a balanced dataset is created by using SMOTE. Hereafter, all four methods are trained on the balanced dataset.

In brief, the results of the first experiment show that the KNN, SVM, RF, and ANN had an F1-score of 0.299, 0.528, 0.338, and 0.451, respectively. This indicates that the ANN outperformed the KNN and RF in the case of imbalanced classes. However, the ANN did not outperform the SVM. The results of the second experiment show that the KNN, SVM, RF, and ANN achieved an F1-score of 0.926, 0.906, 0.929, and 0.899, respectively. These results show that although close, the ANNs did not outperform the traditional methods in the case of balanced classes. Therefore, overall, the findings show that although close, ANNs do not outperform traditional methods in predicting employee attrition.

The contribution of this thesis to the existing literature is twofold. Firstly, this thesis introduces a new method for predicting employee attrition. Secondly, this thesis reveals the relative contribution of neural networks to the task of predicting employee attrition, both in the case of imbalanced and balanced classes.

The remainder of this thesis is as follows. Section 2 discusses Related Work. Section 3 describes the Methods used to predict employee attrition. Section 4 elaborates on the Experimental Setup. Section 5 shows the Results, and Section 6 provides the Discussion and Conclusion.

2. Related Work

2.1 Employee attrition

Employee attrition (employee churn or employee turnover) refers to employees leaving the company over time. Employee attrition can happen either on a voluntary basis or on a non-voluntary basis. Voluntary attrition occurs when good performing employees willingly choose to leave the company for their personal reasons. Non-voluntary attrition, on the other hand, occurs when the company has terminated the contract with an employee for several reasons (Yiğit & Shourabizadeh, 2017; Alduayj & Rajpoot, 2018; Khera & Divya, 2019). Most of the time, a company focuses on voluntary attrition because high-level skilled employees thrive the success of a company (Das & Baruah, 2013; Yiğit & Shourabizadeh, 2017; Alduayj & Rajpoot, 2018). As mentioned in Section 1, when these employees decide to leave, severe drawbacks may occur for the company.

Several studies have examined the causes of employee attrition and found that voluntary employee attrition is determined by more than just one factor (Das & Baruah, 2013; Khera & Divya, 2019). The most common factors that lead to an employee leaving the company relate to job satisfaction. These factors are salary, work environment, work-life balance, recognition, influence in the decision-making process, training options, growth options, and job safety (Das & Baruah, 2013; Ajit & Punnoose, 2016; Khera & Divya, 2019). Furthermore, demographic factors like age, education, gender, ethnic background, and marital status affect an employee's decision to leave or not leave the company (Ajit & Punnoose, 2016). Lastly, the role and behavior of the manager of the employee are among the main reasons for employee attrition (Reina, Rogers, Peterson, Byron, & Hom, 2018). All these factors will be taken into account when predicting employee attrition.

2.2 Previously used machine learning methods in predicting employee attrition

This Section will discuss the results of several studies that have investigated employee attrition predicted by different machine learning methods.

Yiğit & Shourabizadeh (2017) trained several methods for employee attrition ranging in complexity from less complicated to more complicated (Logistic Regression, Naive Bayes, K-Nearest Neighbor (KNN), Decision Trees, Support Vector Machines (SVM), and Random Forests (RF)). In their research, two experiments were conducted, one without feature selection and one with feature selection. The results of both experiments showed that an SVM is the best method for predicting employee attrition.

On the other hand, Ajit & Punnoose (2016) examined the performance of XGBoost compared to other machine learning methods in predicting employee attrition. According to Ajit & Punnoose (2016), the HR-data to predict employee attrition contains noise. The noise can be interpreted by the model as relevant data instead of noise. Therefore, the model learns noise which can result in a model that is not generalizable (Ying, 2019). In contrast to the previously tested models, Ajit & Punnoose

(2016) argued that XGBoost can overcome these problems with noise in the HR-data. The data used in their study is from a global retailer and consists of 73115 rows and 33 columns. Again, several machine learning methods were examined and compared (Logistic Regression, Naive Bayes, Random Forest, KNN, Linear Discriminant Analysis (LDA), SVM, XGBoost). The results showed that the XGBoost method had an AUC score of 0.86 on the test set compared to 0.52 for SVM and 0.51 for a Random Forest. Therefore, the authors suggested that XGBoost is the method that best predicts employee attrition (Ajit & Punnoose, 2016).

Similarly, Jain & Nayyar (2018) argue that XGBoost is a robust method that can handle noise issues in the data. Therefore, this method is generalizable and will result in a high accuracy score. In their research, they evaluate two methods: a Decision Tree and XGBoost. When comparing the results, the XGBoost had an accuracy score of 0.89, and the Decision Tree had an accuracy score of 0.83. Therefore, Jain & Nayyar (2018) argue that XGBoost is the best method to predict employee attrition.

Although present, the abovementioned studies did not take the class imbalance into account. Imbalanced classes cause machine learning methods to ignore the minority class and focus on the majority class, which will result in inaccurate predictions on the minority class (Guo, Yin, Dong, Yang, & Zhou, 2008; Zhu, Baesens, & van den Broucke, 2017; Patel et al., 2020). Therefore, in contrast to the previous studies, Alduayj & Rajpoot (2018) investigated the role of imbalanced classes for predicting employee attrition. Their research focused on three experiments, all with and without feature selection. The first experiment trained several machine learning methods (SVM with different kernel functions, Random Forest, KNN) on the original data. Secondly, the authors trained the same methods on a new dataset created by the adaptive synthetic (ADASYN) approach. ADASYN was used to solve the issue of class imbalances (Alduayj & Rajpoot, 2018). Lastly, Alduayj & Rajpoot (2018) created equal classes in the data by manually undersampling. The primary evaluation metric used in this study was the F1-score. Besides, Accuracy, Precision, and Recall were compared. The first experiment showed that the best method was an SVM with a quadratic kernel, which had an accuracy score of 0.871 and an F1-score of 0.503. Secondly, the results of the ADASYN experiment showed an increase in the performance of all methods. The second experiment's highest performances were for the KNN ($k = 3$) with an F1-score of 0.931, the SVM with a cubic kernel with an F1-score of 0.927, and the Random Forest with an F1-score of 0.921. In the last experiment, information was lost because of the used technique, which led to worse results than the second experiment. However, the SVM with a quadratic kernel obtained an F1-score of 0.740 and an accuracy score of 0.747. In conclusion, the researchers argued that overcoming the issue of imbalanced classes by oversampling will lead to the best results. Furthermore, feature selection did not increase the models' performance in any of the experiments (Alduayj & Rajpoot, 2018).

2.3 Neural Networks

Artificial neural networks (ANNs) are inspired by the structure of the brain's biological neurons and can learn from experience (Jeatrakul & Wong, 2009; Tsai & Lu, 2009; Keramati et al., 2014). Therefore, ANNs provide an output based on knowledge created by learning from the input data (Jeatrakul & Wong, 2009). According to Jeatrakul & Wong (2009) and Tsai & Lu (2009), ANNs in binary classification generate relatively high accuracy scores. Another advantage of an ANN is that it is a non-linear technique that can be applied to most machine learning problems (Jeatrakul & Wong, 2009).

Furthermore, deep learning, which is the use of deep artificial neural networks, i.e., ANNs with many intermediate hidden layers, is applied to many other domains, such as image recognition, object detection, and natural-language processing (Guo, Liu, Oerlemans, Lao, Wu, & Lew, 2016). The application of deep neural networks in the aforementioned domains resulted in several breakthroughs in artificial intelligence. For example, in the domain of object detection, where the Deformable Part Model (DPM) was the best performing model. Nowadays, Convolutional Neural Networks (CNNs) outperform the DPM (Guo et al., 2016).

Despite the advantages of ANNs, breakthroughs of deep learning, and the suggestion of Ajit & Punnoose (2016) to explore neural networks for predicting employee attrition, previous studies on employee attrition prediction have not yet investigated the performance of ANNs on the task. Instead, neural networks have been studied in the context of the closely related task of predicting customer churn (Tsai & Lu, 2009; Keramati et al., 2014). Although employee attrition and customer churn are not identical, they share some similarities (Yiğit & Shourabizadeh, 2017). Because of the abovementioned reasons, it is worth considering the performances of neural networks in customer churn prediction.

Keramati et al. (2014) examined four machine learning methods (Decision Tree, KNN, SVM, ANN) in predicting customer churn in the telecommunication business. For each model, the hyperparameters were tuned to optimize the prediction performance. For example, the number of hidden layers for the ANN, the value of k for KNN, and kernel type for the SVM. The results showed that an ANN is the best performing method with an average F1-score of 0.86. The Decision Tree, KNN, and SVM scored 0.84, 0.75, and 0.83, respectively (Keramati et al., 2014).

Similarly, Tsai & Lu (2009) researched the performance of two types of methods, ANN and hybrid models. Hybrid models are models that combine at least two machine learning models. Tsai & Lu (2009) combined two ANNs for predicting customer churn, where the first ANN was used for data reduction. All models used hyperparameter optimization. Their results showed that the worst-performing ANN had an accuracy score of 0.88 in predicting customer churn. Whereas the best performing ANN had an accuracy score of 0.93. The expanded hybrid model performed even better, with an accuracy score of 0.94.

In conclusion, previous studies on churn prediction suggest that ANNs contribute to the prediction performance. This suggests that it is worthwhile to study the performance of ANNs in

employee attrition. Therefore, this thesis will evaluate the performance of ANNs on the prediction of employee attrition, as suggested by Ajit & Punnoose (2016).

3. Methods

This Section discusses the machine learning methods that will be used for employee attrition prediction. To compare the performance of Artificial Neural Networks to traditional machine learning methods in predicting employee attrition, the following methods are examined: 1) K-Nearest Neighbor, 2) Support Vector Machines, 3) Random Forest, and 4) Artificial Neural Networks. The first three methods are traditional machine learning methods. Furthermore, these methods are used in the research of Alduayj & Rajpoot (2018), where this thesis builds upon. The latter method is a newly introduced method for predicting employee attrition.

3.1 K-Nearest Neighbor (KNN)

KNN is one of the most straightforward methods for binary classification tasks (Alduayj & Rajpoot, 2018). The KNN method classifies a data point based on its k-nearest neighbors, where k is the number of nearest neighboring instances to take into account (James, Witten, Hastie, & Tibshirani, 2013). To determine the closest k instances, a distance measure is used. A commonly used distance measure is the Euclidean distance (Keramati et al., 2014; Ajit & Punnoose, 2016). After the distance measure and the size of k have been chosen, the method classifies unlabeled instances by the majority class of the k-nearest neighbors (James et al., 2013; Keramati et al., 2014; Ajit & Punnoose, 2016).

Figure 3.1 illustrates how KNN works for k = 3. In this Figure, the method is determining the class of the black data point, which can either be purple or yellow. First, the KNN will detect the three data points that are the closest to the black data point. The three closest points are enclosed by the circle in Figure 3.1 and consist of two purple points and one yellow point. Hence, the majority of the data points (2/3) is purple. Therefore, the KNN will decide that the class of the black data point should be purple.

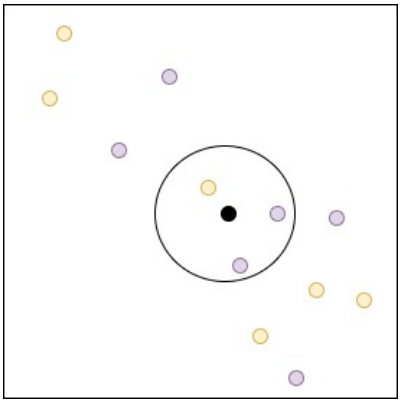

Figure 3.1- *Illustration KNN Approach*

3.2 Support Vector Machines (SVMs)

SVMs are one of the best-performing methods for binary classification tasks and are able to solve linear as well as non-linear classification problems (James et al., 2013; Ajit & Punnoose, 2016). To classify data points, an SVM creates a hyperplane that separates a high-dimensional space into two spaces for each class. Each new space then belongs to a different class (James et al., 2013; Ajit & Punnoose 2016). First, SVMs for linear classification problems will be discussed.

An SVM can be called linear if the hyperplane that separates the classes is a straight line (James et al., 2013). In order to determine the best linearly separable hyperplane out of multiple options, the aim is to maximize the margin. Optimizing the margin means maximizing the distance between the closest data points, also known as support vectors, on both sides of the hyperplane. Resulting in a hyperplane that is as far as possible from the closest data points (support vectors). In short, the support vectors can be seen as parallel hyperplanes to the main hyperplane that help create the optimal hyperplane (James et al., 2013; Keramati et al., 2014). The left panel of Figure 3.2 provides an illustration of choosing the best hyperplane for a linear classification task by maximizing the distance between the support vectors (dotted parallel lines) and the hyperplane.

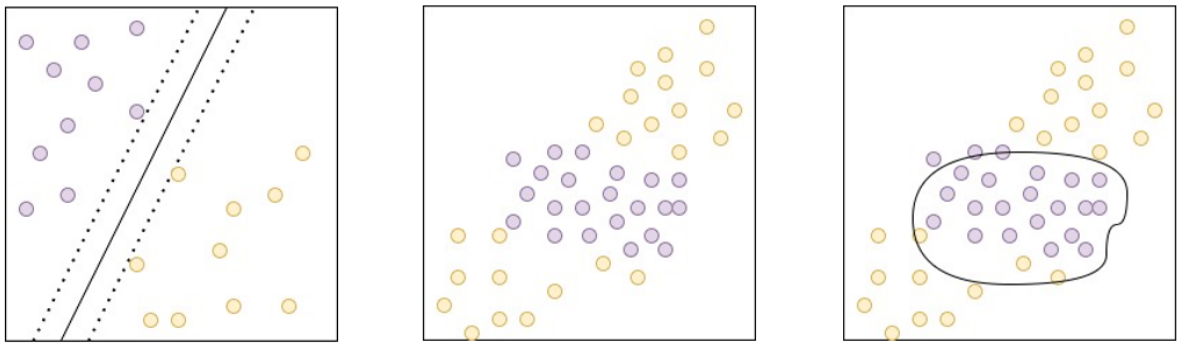

Figure 3.2- *Left: Illustration linear SVM approach. Middle: mixed purple and yellow data points. Right: Illustration rbf SVM approach.*

However, not all classification tasks are linearly separable. For example, when the purple and yellow dots are divided through the feature space as shown in the middle panel of Figure 3.2. In the middle panel of Figure 3.2 no straight hyperplane can separate the feature space into two in order to correctly separate the classes. To solve such a non-linear classification task, a kernel function can be applied. The kernel function returns the dot product of two vectors which augments the feature space. In such a way, an SVM can create a non-linear hyperplane (James et al., 2013; Alduayj & Rajpoot, 2018). Several non-linear kernel functions can be applied, such as polynomial, sigmoid, and radial basis function (rbf) (James et al., 2013; Keramati et al., 2014; Alduayj & Rajpoot, 2018). The right panel of Figure 3.2 illustrates how the rbf kernel would separate the space.

3.3 Random Forests (RFs)

Random Forests is a method that uses Decision Trees as building blocks to create a prediction (James et al., 2013). A RF is a method where multiple trees are created on a subset of bootstrapped training samples (Ajit & Punnoose, 2016). After a specified number of trees (n) are created, a majority vote is taken to make the final prediction. Each time there is a split, a random sample of predictors (m) is chosen out of the full sample of predictors (p). In this way, the trees will be decorrelated, which results in less variance. By reducing the variance and leaving the bias unchanged, a Random Forest is able to improve the performance of a single Decision Tree (James et al., 2013; Ajit & Punnoose, 2016).

3.4 Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs) are increasingly used for classification as the use of the networks results in relatively high accuracy scores (Jeatrakul & Wong, 2009; Tsai & Lu, 2009). The structure and function of an ANN is loosely based on the structure and function of neural networks in the human brain (Gershenson, 2003; Keramati et al., 2014). When a neuron in the human brain receives a signal that is strong enough, the neuron will be activated. If a neuron is activated it passes on a signal (output of the neuron), which can be an input for another neuron (Gershenson, 2003). ANNs work in a similar manner. Figure 3.3 shows a basic form of an ANN that consists of an input layer, at least one hidden layer, and an output layer. Typically, the prediction power of ANNs increases with the number of hidden layers, although the associated increase in connection weights (the adaptable parameters of the ANN), requires an increase in the size of the dataset.

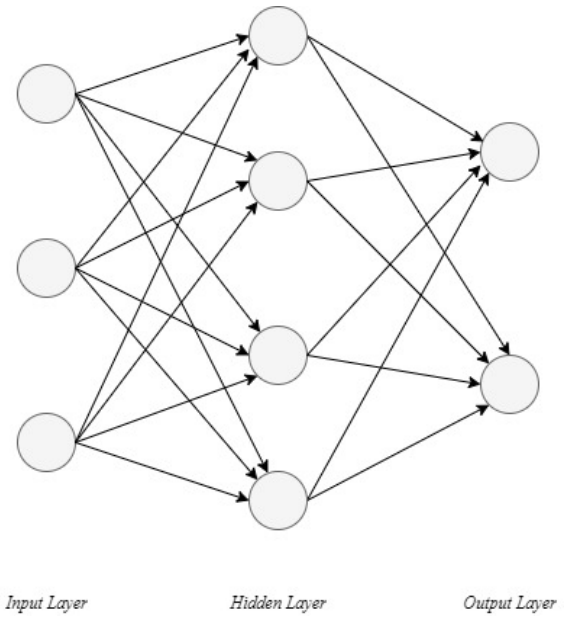

Figure 3.3 - *Illustration of an artificial neural network with one hidden layer.*

When training an ANN as presented in Figure 3.3, the data is inserted into the input layer. Hereafter, information is fed forward to the hidden layer. Therefore, this type of neural network is also known as a feedforward neural network. During the process of feeding information forward, the

information is multiplied by weights plus a bias. The value of this multiplication is fed through an activation function in the node, which decides if a node is activated. The activation function in a node is a non-linear transformation of the input that goes into the node. The outcome of this activation is then forwarded to the output layer. Again during this process, the value is multiplied by weights plus a bias. Hereafter the activation function of the output node determines the output value of the input. The activation function in the output node creates probabilities, after which the value with the highest probability is chosen (Gershenson, 2003; Jeatrakul & Wong, 2009).

To evaluate the ANN during training, the predicted output of the ANN will be compared to the actual output. The difference between these outputs results in an error. To minimize the error, the error will be sent back through the network, where every weight and bias of all nodes will be updated accordingly. This phenomenon is called backpropagation. Backpropagation starts at the output layer and moves back to the input layer. When the training of an ANN starts, weights and biases are set randomly. The goal of backpropagation is to adjust the weights and biases in such a way that the error is minimized. (Gershenson, 2003; Jeatrakul & Wong, 2009). The process of backpropagation will be repeated many times, until the weights and biases result in an error that is acceptable. The weights and biases can be seen as knowledge that is learned in the network (Jeatrakul & Wong, 2009).

After the error is minimized, the network is ready to receive new, unseen, data. When new data is entered into the network, the network will generate predictions based on the knowledge that is learned during the training phase (Jeatrakul & Wong, 2009).

4. Experimental Setup

This Section discusses the dataset used for our research, followed by the exploration, preprocessing, and cleaning steps. Subsequently, all methods and parameter tuning for all the methods performed in this thesis are discussed. Lastly, this Section elaborates on the evaluation metrics that are used to evaluate the performance of the methods. All programming steps discussed in this Section are implemented by using Python.

4.1 Dataset Description

The dataset used to answer the research question is published by IBM Watson Analytics on Kaggle.com and is publicly available in CSV format. The dataset contains information that is HR-related and exists of 1470 rows and 35 columns. The target variable in the dataset is called "Attrition" and is divided into two groups, "Yes" and "No". Where "Yes" means that an employee has left the company and "No" means that an employee is still with the company. The target variable is imbalanced because only 237 rows contain "Yes", whereas 1233 rows have "No". The other features in the dataset are related to the literature of Section 2.1 and are listed in Table A1 of the Appendix.

4.2 Data Preprocessing

For the methods to be able to handle data well, the dataset first needs to be explored, preprocessed, and cleaned. The exploration, cleaning, and preprocessing steps will be discussed in this Section.

We dropped the variables "Standard Hours" and "Employee Count", which is in accordance with the research of Alduayj & Rajpoot (2018). In addition, we disregard the variables "Employee Number", "Over 18", and "MonthlyIncome". "Employee Number" will be dropped because it does not indicate something special about the employee and the variable does not relate to the literature discussed in Section 2.1. The variable "Over18" will be dropped because everyone in the dataset is older than 18. Lastly, the variable "MonthlyIncome" will be dropped because it measures the same as "MonthlyRate". Therefore, the remaining dataset will consist of 1470 rows and 30 columns. After dropping the unnecessary variables, the remaining dataset is checked for missing values, odd values, and outliers. No missing values and odd values were found. To identify outliers, boxplots of variables are created that revealed outliers for the variables "NumCompaniesWorked", "TrainingTimesLastYear", "YearsAtCompany", "TotalWorkingYears", "YearsInCurrentRole", "YearsSinceLastPromotion", "YearsWithCurrentManager". Closer examination revealed that these outliers are not likely to be incorrect values or values because of incorrect interpretation. For example, one outlier is someone who has been with the same manager for 15 years. This person also has worked at the company for 17 years, which makes the 15 years with the same manager not unlikely. For this reason, the outliers are not removed from the dataset or replaced by other values. Figure A1 in the Appendix shows the boxplots of all variables.

Subsequently, the data are split into input features and a target variable. The target variable is "Attrition" and the input features are all the remaining variables. "Attrition" has two values, "Yes" and "No". However, machine learning methods can only process numeric target values. Therefore, LabelEncoder from the preprocessing library of scikit-learn in Python is used to transform the target variable into binary values (Pedregosa et al., 2011). Furthermore, the nominal input features are transformed into numeric values since most methods cannot work with categorical input data directly. Because nominal categorical features do not have a ranking, one-hot encoding is used to created the numeric values. One-hot encoding is implemented with the get_dummies library form pandas (Pedregosa et al., 2011). Table A.1 in the Appendix shows which features are nominal categorical features. Next, a balanced dataset needs to be created in order to answer research question 1.b. Synthetic Minority Oversampling Technique (SMOTE) is used to balance the imbalanced dataset. SMOTE balances the imbalanced dataset by creating new minority records based on a combination of existing minority records. (Chawla, Bowyer, Hall & Kegelmeyer, 2002; Patel et al., 2020). The Python package used to apply SMOTE is SMOTE from imblearn over_sampling (Pedregosa et al., 2011). SMOTE changes the frequency of occurrence of leavers, which deviates from the frequency of occurrence of leavers in real life. This might have consequences for the generalizability of the methods. However, performing SMOTE is likely to overcome the methods from biasing towards the majority class (Guo et al., 2008). Therefore, it is beneficial to perform.

Both the imbalanced and balanced datasets are split into stratified training and test sets using the train_test_split from the model_selection library of scikit-learn in Python (Pedregosa et al., 2011). Since the original dataset is imbalanced, a stratified split is used to make sure that the ratio of 'leavers' is the same among training and test sets. Both training sets contain 80% of the data and both test sets contain the remaining 20% of the data. A random seed is set in order for the results to be reproducible.

The last preprocessing step is normalizing all input datasets because some methods, for example KNNs, are highly sensitive to scale (Ajit & Punnoose, 2016). In order to be consistent, normalization is applied to the input data for all methods. To implement normalization the MinMaxScaler from the preprocessing library of scikit-learn is used, which scales the data in a range of [0,1] (Pedregosa et al., 2011).

4.3 Algorithms and Hyperparameter Tuning

This research examines to what extent neural networks can outperform traditional machine learning methods in employee attrition prediction. As mentioned in Section 3, to perform this binary classification task, KNN, SVM, RF, and ANNs will be examined. In order to answer the research question, two experiments are carried out. All methods and their parameter settings are examined on the imbalanced and balanced datasets. The supervised classification methods and their parameters will be discussed below. To implement hyperparameter tuning and 10-fold cross-validation GridSearchCV

from the scikit-learn model_selection library is used for all methods (Pedregosa et al., 2011). Different hyperparameters are tuned for different methods. To make sure that the results are reproducible, a random seed is set during the tuning of the methods.

First, for KNN, the KNeighborsClassifier module from scikit-learn is used (Pedregosa et al., 2011). As the value of k has an impact on the decision boundary this hyperparameter is optimized (James et al., 2013). Different values of k ranging from 3 to 19 in steps of 2 are examined. The values of k are chosen to be odd so that it is not possible to get a tie in a majority vote (Keramati et al., 2014). Furthermore, the value k = 1 is not considered because it has a high chance of overfitting, since the classification is only based on the 1 closest neighbor (James et al., 2013). When k is too large the decision boundary can become close to linear and might underfit the data (James et al., 2013). Therefore, 19 is chosen as the maximum number of neighbors. The distance measure used is the Euclidean distance, since this is the most commonly used distance measure for KNN (Keramati et al., 2014).

Second, to implement the SVM, the SVC module form scikit-learn is used (Pedregosa et al., 2011). As mentioned in Section 3.2, SVMs can have linear or non-linear kernels. Therefore, the following kernel types are tuned: linear, cubic polynomial, quadratic polynomial, rbf, and sigmoid. In addition, different values of C are tuned. C is a parameter that determines how many errors the method is allowed to make. When C is small the method is not allowed to have a margin that is highly violated, when C is large the method is allowed to have a margin that has more violations. In this way C relates to under and overfitting (James et al., 2013). The values of C that are examined are 0.001, 0.01, 0.1, 1, and 10. Lastly, when the rbf, poly, and sigmoid kernels are applied the parameter gamma is tuned. Gamma affects the complexity of the decision boundary. When gamma is high there is more curvature, and when gamma is small there is less curvature (James et al., 2013). The values of gamma that are examined are 0.001, 0.01, 0.1, 1, 10.

Third, for RF, the RandomForestClassifier module from scikit-learn in used (Pedregosa et al., 2011). The following hyperparameters are tuned: the number of trees to average over, the maximum depth of the trees and the maximum number of features to choose from when making a split. The values that are examined for the number of trees to average over are 10, 20, 50, 100 and 200. For the maximum depth of the trees, the values tested range from 5 to 30 in steps of 5. To prevent overfitting the value of 30 is chosen as a maximum. Because the deeper a tree is, the more likely it will overfit, especially in a small dataset like the one used in this research. For the number of features to choose from when making a split the values 5, 6, 7, 8, and 9 are examined. The default for this parameter is the square root of the number of input features. The square root of the number of input features in this research is 7. Therefore, 7 is chosen as the center. Some values above and below 7 are chosen to see if they perform petter than the default.

Lastly, for the ANNs the Sequential model from the library Keras is used (Chollet & others, 2015). This research trained several ANNs with different depths. As stated in Section 3.4, the prediction performance of neural networks often increases with its depth (Guo et al., 2016; Telgarsky, 2016), but

brings a cost in terms of the dataset size. The dataset used in this thesis is small. Therefore, the neural networks trained on this dataset cannot be too deep. Transfer learning might be an option to create deeper neural networks on a small dataset. When transfer learning is applied, newly trained neural networks use feature representations of an already learned neural network for a similar task. In this way 'knowledge' is transferred (Brownlee, 2017). However, transfer learning will only be successful if the feature representations used are generalizable (Brownlee, 2017). Since the input data of customer churn and employee attrition is different, the learned feature representations are likely to be different and not generalizable. Therefore, the application of transfer learning in predicting employee attrition is not possible. With the small dataset in mind, several depths will be explored, starting with 1, 2, and 3 hidden layers. Hereafter a jump to 5 is made to see if the performance significantly improves. Furthermore, the number of hidden nodes is tuned. The values for the number of hidden nodes are between ⅔ of the size of the input layer and the size of the output layer. Some random numbers are chosen with trial and error, because not all values can be examined due to the limited computational resources at our disposal. The starting values of the nodes in the hidden layer that are examined range from 15 to 33. For the second hidden layer the starting values of the nodes that are examined range from 8 to 25. The starting values of the nodes of the third hidden layer range from 6 to 23. The fourth and fifth hidden layer node values that are examined range from 3 to 12 in steps of 3. Besides, batch sizes of 32, 64, and 128 are tuned. As an optimizer Adam is used because it combines the advantages of RMSprop and Adagrad. Furthermore, Adam is most commonly used in training neural networks (Bock & Weiß, 2019).

4.4 Evaluation

Accuracy is a frequently used evaluation metric in classification problems. Accuracy calculates the percentage of all the correct predicted instances (Guo et al., 2008; Hossin & Sulaiman, 2015). In this case, Accuracy is the percentage of leavers classified as leavers (true positives) and non-leavers classified as non-leavers (true negatives). However, in the case of imbalanced classes, the dataset consists of more non-leavers than leavers. Therefore, the methods have a bias towards the majority class (non-leavers), which will result in a high true negative rate. As a result, the overall accuracy score is high while the methods do not perform well on the minority class (leavers) (Guo et al., 2008). For this reason, this research uses another evaluation metric, the F1-score.

The F1-score is a frequently used measure in imbalanced classification tasks (Guo et al., 2008; Luo, Pan, Wang, Ye, & Qian, 2019). The formula of the F1- score is as follows:

$$\boldsymbol{F1 - score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (\mathbf{1})$$

As shown above, F1-score is a mixture of Precision and recall, which are powerful measures in imbalanced classification tasks (Guo et al., 2008; Luo et al., 2019). Furthermore, Alduayj & Rajpoot (2018) used the F1-score as an evaluation metric for their methods because they took the issue of imbalanced classes into account. As a result, the main evaluation metric for the methods in this research

is the F1-score. In addition, Accuracy, Recall, and Precision will be reported. Accuracy is reported to gain insight into the relationship between Accuracy and the F1-score in the case of imbalanced classes. Recall and Precision are reported to get a full understanding of the methods' performance.

The baseline methods for this research are SVM, KNN, and RF, which are based on the research of Alduayj & Rajpoot (2018). After the best hyperparameters are found for all methods, these methods are trained on the complete training set. To find the final F1-score for the different methods, all methods will be applied on the test set. In order to determine whether ANNs outperform the previously mentioned methods, the final F1-score of the ANN will be compared to the final F1-score of the baseline methods.

Lastly, to create visualizations of the methods' performance, the 'matplotlib' and 'seaborn' libraries in Python are used (Hunter, 2007; Waskom, et al., 2017). The link to the full Python script can be found in the Appendix.

5. Results

This Section presents the results of the two experiments described in Section 4.3. Section 5.1 shows the performances of the algorithms on the imbalanced dataset. Section 5.2 reports the performances of the algorithms on the balanced dataset.

5.1 Performance of Algorithms on Imbalanced Data

Algorithm	Accuracy	F1	Precision	Recall
KNN	0.840	0.299	0.500	0.213
SVM	**0.884**	**0.528**	**0.760**	**0.404**
RF	0.854	0.338	0.611	0.234
ANN	0.867	0.451	0.667	0.340

Table 5.1- *Performance of Algorithms on Imbalanced Data*

Table 5.1 displays the performances of the best performing KNN, SVM, RF, and ANN on the imbalanced test set. The best parameter for the KNN is: (k = 5). For the SVM the best parameters are: (kernel = linear, C = 1). The best parameters for the RF are: (max_depth = 10, max_features = 8, n_estimators = 10). Lastly, the best parameters for the ANN are: (number of hidden layers = 3, batch_size = 64, epochs = 15, hidden_nodes_1 = 26, hidden_nodes_2 = 25, hidden_nodes_3 = 15).

The results of most of the baseline methods (KNN, SVM, RF) are in line with the results of the methods in the research of Alduayj & Rajpoot (2018). As Table 5.1 shows, the best F1-score (0.528) is achieved by the SVM. Similarly, in the research of Alduayj & Rajput (2018), the best performing method was an SVM, which had an F1-score of 0.503. However, the performance of the KNN in this research significantly differs from the performance of the KNN in the research of Alduayj & Rajput (2018). Their best performing KNN had an F1-score of 0.08, whereas the F1-score of the KNN in this research is 0.299. Since KNN is sensitive to scale, the difference may be caused by normalization of the data. In this research we normalized the data, whereas Alduayj & Rajpoot (2018) did not normalize the data. To verify if normalization indeed caused the difference, we additionally ran a KNN on the imbalanced data that is not normalized. Table 5.2 shows the results of this experiment.

Algorithm	Accuracy	F1	Precision	Recall
KNN	0.820	0.185	0.333	0.128

Table 5.2- *Performance of KNN on not normalized Imbalanced Data*

Table 5.2 displays that the KNN on not normalized data has an F1-score of 0.185 and an Accuracy score of 0.82. The F1-score of the KNN on not normalized data (0.185) is lower than the F1-score of the KNN on normalized data (0.299). This indicates that normalization indeed benefits the performance of the KNN and might cause the difference in results. However, the F1-score of the KNN

on not normalized data is still higher than the results of the KNN in the research of Alduayj & Rajput (2018). The remaining difference might be caused by other different preprocessing steps. For example, they gave the nominal features a numeric value ranging from 1 to 3, whereas this research applied one-hot encoding on the nominal features. Furthermore, this research dropped more features than the research of Alduayj & Rajput (2018), which changes the feature space.

Moreover, Table 5.1 shows that the ANN outperforms the KNN and the RF. The F1-scores are 0.451 versus 0.299 and 0.338 respectively. Meanwhile, the SVM achieves an F1-score of 0.528. Although close, the ANN does not outperform the SVM. Therefore, in the case of imbalanced classes, the ANN does not outperform all traditional machine learning methods.

In addition, Table 5.1 shows that the Accuracy score of all methods is quite high while the F1-score is relatively low. These results are in line with the results of Alduayj & Rajpoot (2018). When Accuracy is high and the F1-score is relatively low, this indicates that the method ignores the minority class and focuses on the majority class. The results show that this especially holds for KNN and RF. The confusion matrices (CM) of all methods in Figure 5.1 further illustrate the ignorance of the minority class. The top left CM in Figure 5.1 presents the results of the KNN. Although there are 47 leavers in the dataset, the KNN only predicts 10 out of 47 as leavers. For the SVM (top right of Figure 5.1) this is 19 out of 47, for the RF (bottom left of Figure 5.1) 11 out of 47, and for the ANN (bottom right of Figure 5.1) 16 out of 47.

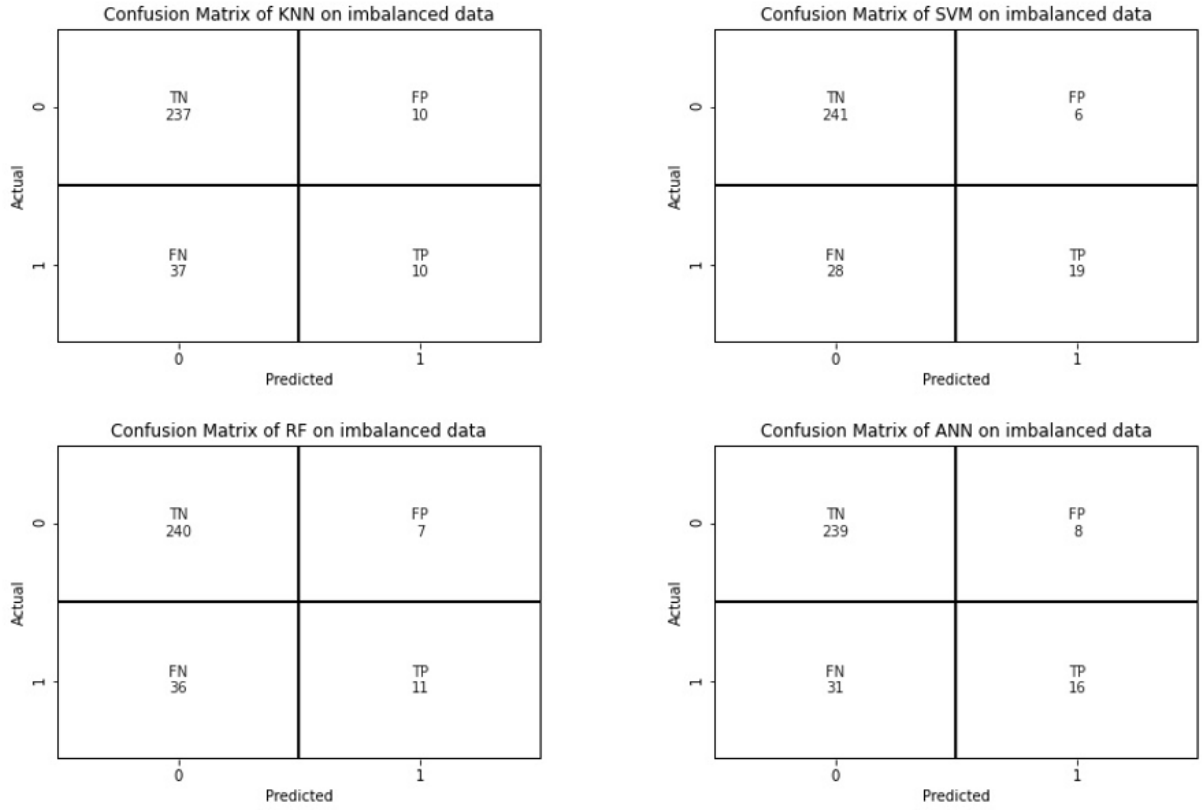

Figure 5.1- *Confusion Matrices of all Algorithms on Imbalanced Data.*

5.2 Performance of Algorithms on Balanced Data

Algorithm	Accuracy	F1	Precision	Recall
KNN	0.927	0.926	0.945	**0.907**
SVM	0.911	0.906	0.963	0.854
RF	**0.933**	**0.929**	**0.991**	0.874
ANN	0.905	0.899	0.955	0.850

Table 5.3- *Performance of Algorithms on Balanced Data.*

Table 5.3 displays the results of the best performing KNN, SVM, RF, and ANN on the balanced test set. The best parameter for the KNN is: (k = 3). For the SVM the best parameters are: (kernel = linear, C = 10). For the RF the best parameters are: (max_depth = 20, max_features = 7, n_estimators = 200). Lastly, the best parameters for the ANN are: (number of hidden layers = 3, batch_size = 32, epochs = 15, hidden_nodes_1 = 24, hidden_nodes_2 = 14, hidden_nodes_3 = 9).

The results of the baseline methods KNN, SVM, and RF on balanced data in Table 5.3 are close to the results obtained by Alduayj & Rajpoot (2018). Nonetheless there are some slight differences. For example, their best method was KNN (k = 1) with an F1-score of 0.967. However, as Table 5.3 shows, the best performing baseline method in this research is the RF with an F1-score of 0.929. Despite the difference in the best performing method, the F1-score of the RF in the research of Alduayj & Rajpoot (2018) is close to the F1-score obtained in this thesis. Namely, 0.921 versus 0.929. The difference in the performance of KNN may arise because this thesis has not examined k = 1 due to the chance of overfitting. However, the second-best method in the research of Alduayj & Rajpoot (2018) is KNN with k = 3, which had an F1-score of 0.931. The performance of their second best KNN is close to the performance of the KNN in this research (F1-score = 0.926). As described above, there are some slight differences with the results of Alduayj & Rajpoot (2018). A possible explanation for these differences is that this research uses different preprocessing steps. Furthermore, the slight differences can arise because the balanced dataset of Alduayj & Rajpoot (2018) contained 1152 leavers and 1233 non-leavers. Whereas, in this research both classes were equally distributed, with 1233 rows for each class.

With regards to the performance of the ANN, Table 5.3 shows that the ANN has an F1-score of 0.899. However, although close, Table 5.3 indicates that the F1- score of the ANN does not exceed the F1-score of the KNN (0.926), SVM (0.906), and RF (0.929). Therefore, on a balanced dataset the ANN does not outperform the traditional methods in predicting employee attrition.

Furthermore, unlike the results in the previous Section, Table 5.3 shows that both the Accuracy and F1-score are relatively high. This indicates that the methods do not ignore a class anymore. Figure 5.2 shows the confusion matrices of all methods on the balanced data and further illustrates that the methods do not ignore a class anymore. Figure 5.2 shows that the majority of leavers is correctly predicted by all methods. For example, the KNN (top left of Figure 5.2) predicts 224 out of the 247

leavers correct, the SVM (top right of Figure 5.2) 211 out of 247, the RF (bottom left of Figure 5.2) 216 out of 247, and the ANN (bottom right of Figure 5.2) 210 out of 247.

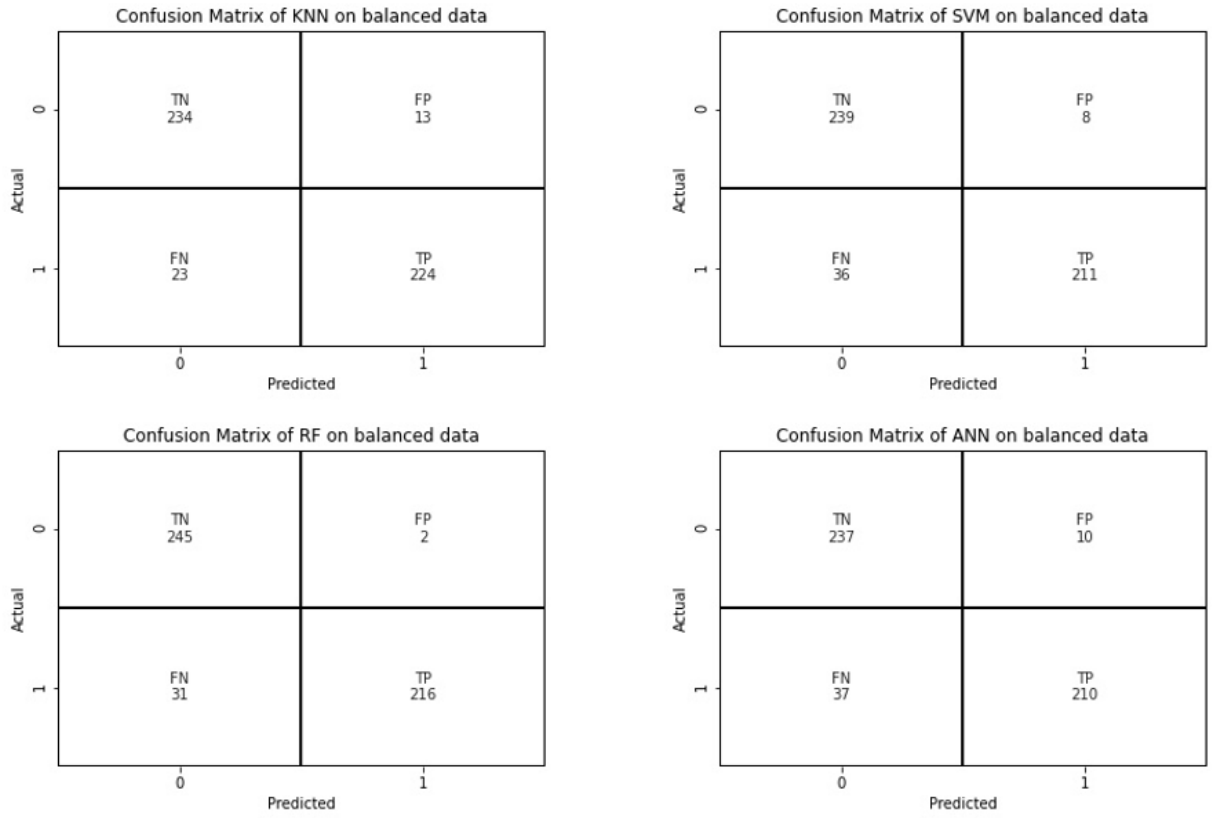

Figure 5.2- *Confusion Matrices of all Algorithms on Balanced Data.*

6. Discussion and Conclusion

6.1 Goal and Findings

The goal of this thesis was to examine whether and to what extent neural networks outperform traditional machine learning methods in predicting employee attrition. Employee attrition can cause several drawbacks for a company, such as additional costs that arise and harm to the reputation of the company (Alduayj & Rajpoot, 2018; Frye et al., 2018; Khera & Divya, 2019). Therefore, employee attrition is a rising topic of interest for a company. In order to reduce the drawbacks, companies want to know which employee is likely to leave. Machine learning methods can be used to predict which employee is likely to leave. With that information, the company might be able to prevent the employees from leaving (Ajit & Punnoose, 2016; Yiğit & Shourabizadeh, 2017).

Employee attrition is a rare event which results in imbalanced classes. Imbalanced classes can cause the algorithm to ignore the minority class and focus on the majority class (Guo et al., 2008; Zhu et al., 2017; Patel et al., 2020). To take this effect into account, the main research question is split up into two sub questions. First: 'To what extent do neural networks outperform traditional machine learning methods in predicting employee attrition when the dataset is imbalanced?'. Second: 'To what extent do neural networks outperform traditional machine learning methods in predicting employee attrition when the dataset is balanced?'

Accordingly, this research carried out two experiments. First, four methods (KNN, SVM, RF, and ANNs) were trained on the original dataset with imbalanced classes. Second, this research used SMOTE to create a balanced dataset. Hereafter, KNN, SVM, RF, and ANNs were trained on the balanced dataset. To get the best performing methods, 10-fold cross-validation and hyperparameter tuning was applied in both experiments. As mentioned in Section 4.4, the final best performing methods were evaluated on the F1-score.

The results of the first experiment showed that the ANN scored an F1-score of 0.451, whereas the KNN, SVM, and RF scored 0.299, 0.528, 0.338 respectively. This indicates that in the case of imbalanced classes, the ANN outperformed the KNN and RF. However, although close, the ANN did not outperform the SVM. Therefore, in the case of imbalanced classes, ANNs do not outperform all traditional methods in predicting employee attrition. This result contradicts the findings of Keramati et al. (2014), who found that the ANN outperformed all traditional methods in predicting customer churn. A possible explanation for this difference is that the original dataset in this research very small. In order for ANNs to learn and be generalizable, ANNs need a large amount of data. SVMs on the other hand, are better able to handle small data. Therefore, the SVM can have outperformed the ANN in this first experiment. With regards to KNN, KNN is sensitive to skewness in the target data. If there is a majority in the target data, KNN is likely to predict the new example with the majority class (Liu & Chawla, 2011). This might explain why KNN performs worse than the ANN, SVM, and RF. In addition, the results of the first experiment show that the Accuracy score for all methods is high, while the F1-score is relatively low. This indicates that the methods tend to ignore the minority class and focus on the

majority class in the case of imbalanced classes. This finding is in line with the findings of Alduayj & Rajpoot (2018).

In the case of balanced classes, the ANN achieved an F1-score of 0.899, whereas the KNN, SVM, and RF achieved 0.926, 0.906, and 0.929 respectively. These results indicate that the ANN does not outperform the traditional machine learning methods in predicting employee attrition in the case of balanced classes. Although this finding is not in line with the results of Keramati et al. (2014), there might be a possible explanation for the difference in results. As mentioned above, their research used a larger dataset, whereas this research used a small dataset. Although the balanced dataset is larger than the original dataset it is still small. Simple methods like the traditional methods are better able to learn from small datasets than ANNs. ANNs often need more data to learn generalizable feature representations. Therefore, the ANN in this research might have not been able to achieve its full potential. Furthermore, the research of Keramati et al. (2014) used the original dataset without applying SMOTE. Therefore, the data is distributed differently, which might cause a difference in performance. However, in this research we applied SMOTE. Because of SMOTE, there is no majority target value in the dataset anymore. Therefore, it might be that the KNN in the second experiment is performing better than in the first experiment. With regards to the ignorance of the minority class, the results show that both the Accuracy and F1-score are relatively high. Therefore, the methods do not tend to ignore a class anymore.

In general, although the results of both experiments in this thesis show that performance of the ANNs are close to the performance of the traditional methods, the ANNs do not outperform traditional machine learning methods in predicting employee attrition.

6.2 Implications

This thesis builds on prior research regarding employee attrition prediction. Existing research has no consistent answer to what machine learning method is best in predicting employee attrition. Furthermore, to predict employee attrition, existing research only used traditional machine learning methods (KNN, SVMs, RF) (Ajit & Punnoose, 2016; Yiğit & Shourabizadeh, 2017; Alduayj & Rajpoot, 2018). The performance of ANNs has not been examined in employee attrition prediction. As opposed to customer churn prediction, where research showed that ANNs outperform traditional machine learning methods (Tsai & Lu, 2009; Keramati et al., 2014).

This thesis compares the performance of ANNs to the performance of traditional machine learning methods in predicting employee attrition. Therefore, this thesis extends prior literature regarding employee attrition prediction in two-fold. First, this thesis introduces a new method for predicting employee attrition. Second, this thesis shows the relative contribution of neural networks to the task of predicting employee attrition, both in the case of imbalanced and balanced classes.

6.3 Limitations and Future Research

This thesis contains multiple limitations, which provide several future research areas. The limitations and future research areas are discussed in this Section.

First, due to time and computational constraints, not all hyperparameters for the ANN have been optimized. This might indicate that the optimal hyperparameters for the ANN have not been found. Likewise, not all hyperparameters for the traditional methods have been tuned. Therefore, future research can try to optimize the ANN further. For example, future research can try different numbers of hidden nodes, different optimization functions, extra hidden layers, and different epochs. Next to that, future research can try to implement drop out techniques. In addition to the hyperparameters of the ANN, future research might try to optimize the traditional methods further.

Second, in this thesis small datasets are used to train and evaluate the methods. The original dataset consists of 1470 instances and the balanced dataset contains 2466 instances. This might be too small for the ANN to be able to learn generalizable feature representations. Furthermore, small datasets can have several other limitations. For example, overfitting is more likely to occur when algorithms are trained on a small dataset. To overcome the issues of a small dataset, future research might replicate this study on a large dataset. Furthermore, as the value of neural networks often increases with its depth, future research might try to train a deeper neural network on a large dataset. In addition, when enough time and resources are available, future research might extend the experiments in this research by optimizing the methods further, as mentioned above.

Third, in this thesis we used SMOTE to create a balanced dataset. The balanced dataset is created to solve the ignorance of the minority class by the methods on the original small dataset, like in the research of Alduayj & Rajput (2018). However, SMOTE changes the frequency of occurrence of leavers, which deviates from the frequency of occurrence of leavers in real life. This might affect the generalizability of the methods. Therefore, future research can investigate the role of SMOTE. For example, future research might try a different distribution of the target variable "Attrition' such as, 70% non-leavers and 30% leavers. Or future research might train the methods on a balanced dataset by using SMOTE and evaluate the methods on a test set that is not made balanced.

Lastly, in this thesis we applied normalization on the input features. As shown in Section 5, for KNNs normalization can be beneficial. However, normalization affects the data, which might affect the performance of the methods. Therefore, future research can investigate what the role of normalization is. To verify the effect of normalization, future research can replicate this research with and without normalization and compare the performance of the methods.

References

Ajit, P., & Punnoose, R. (2016). Prediction of Employee Turnover in Organizations using Machine Learning Algorithms. *International Journal of Advanced Research In Artificial Intelligence (IJARAI)*, *5(9)*, pp. 22-26.

Alduayj, S. S., & Rajpoot, K. (2018). Predicting Employee Attrition using Machine Learning. *In 2018 International Conference on Innovations in Information Technology (IIT)* (pp. 93-98). IEEE.

Bock, S., & Weiß, M. (2019). A Proof of Local Convergence for the Adam Optimizer. *In 2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

Brownlee, J. (2017, December 20). *Machine Learning Mastery*. Retrieved from A Gentle Introduction to Transfer Learning for Deep Learning: https://machinelearningmastery.com/transfer-learning-for-deep-learning/

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority oversampling technique. *Journal of artificial intelligence research, 16*, 321-357.

Chollet, F., & & others. (2015). *Keras.Github*. Retrieved from https://github.com/fchollet/keras

Das, B. L., & Baruah, M. (2013). Employee Retention: A Review of Literature. *Journal of Business and Management, 14*(2), pp. 8-16.

Das, R. C., & Devi, S. A. (2020). Conceptualizing the Importance of HR Analytics in Attrition Reduction. *International Research Journal on Advanced Science Hub, 2*(10s), pp. 40-48.

Frye, A., Boomhower, C., Smith, M., Vitovsky, L., & Fabricant, S. (2018). Employee Attrition: What Makes an Employee Quit? *SMU Data Science Review, 1*(1), 9.

Gershenson, C. (2003). Artificial neural networks for beginners. *arXiv preprint cs/0308031*.

Guo, X., Yin, Y., Dong, C., Yang, G., & Zhou, G. (2008). On the Class Imbalance Problem. *In 2008 Fourth international conference on natural computation. 4*, pp. 192-201. IEEE.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing, 187*, 27-48.

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process, 5*(2), 1-11.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering, 9*(3), 90-95.

Jain, R., & Nayyar, A. (2018). Predicting employee attrition using xgboost machine learning approach. *In 2018 International Conference on System Modeling & Advancement in Research Trends (SMART)* (pp. 113-120). IEEE.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning. New York: Springer.

Jeatrakul, P., & Wong, K. W. (2009). Comparing the Performance of Different Neural Networks for Binary Classification Problems. *In 2009 Eighth International Symposium on Natural Language Processing* (pp. 111-115). IEEE.

Keramati, A., Jafari-Marandi, R., Aliannejadi, M., Ahmadian, I., Mozaffari, M., & Abbasi, U. (2014). Improved churn prediction in telecommunication industry using data mining techniques. *Applied Soft Computing, 24*, pp. 994-1012.

Khera, S. N., & Divya. (2019). Predictive Modelling of Employee Turnover in Indian IT Industry Using Machine Learning Techniques. *Vision, 23*(1), pp. 12-21.

Liu, W., & Chawla, S. (2011). Class confidence weighted knn algorithms for imbalanced data sets. *In Pacific-Asia conference on knowledge on knowledge discovery and data mining* (pp. 345-356). Berlin, Heidelberg: Springer.

Luo, H., Pan, X., Wang, Q., Ye, S., & Qian, Y. (2019). Logistic regression and random forest for effective imbalanced classification. *In 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). 1*, pp. 916-917. IEEE.

Patel, H., Rajput, D. S., Reddy, G. T., Iwendi, C., Bashir, A. K., & Jo, O. (2020). A review on classification of imbalanced data for wireless sensor networks. *International Journal of Distributed Sensor Networks, 16*(4), 1-15.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12*, 2825-2830.

Reina, C. S., Rogers, K. M., Peterson, S. J., Byron, K., & Hom, P. W. (3018). Quitting the bos? The Role of Manager Influence Tactics and Employee Emotional Engagement in Voluntary Turnover. *Journal of Leadership & Organizational Studies, 25*(1), 5-18.

Telgarsky, M. (2016). Benefits of depth in neural networks. *In Conference on learning theory* (pp. 1517-1539). PMLR.

Tsai, C. F., & Lu, Y. H. (2009). Customer churn prediction by hybrid neural networks. *Expert Systems, 36*(10), 12547-12553.

Waskom, M., Botvinnik, O. O., Kane, D., Hobson, P., Lukauskas, S., Gemperline, D. C., . . . Qalieh, A. (2017, September). mwaskom/seaborn: v0.8.1. *Zendo*.

Yiğit, I. O., & Shourabizadeh, H. (2017). An Approach for Predicting Employee Churn by Using Data Mining. *In 2017 International Artificial Intelligence and Data Processing Symposium (IDAP)* (pp. 1-4). IEEE.

Ying, X. (n.d.). An overview of overfitting and its solutions. *In Journal of Physics: Conference Series (Vol.1168, No.2).* 022022: IOP Publishing.

Zhu, B., Baesens, B., & van den Broucke, S. K. (2017). An empirical comparison of techniques for the class imbalance problem in churn prediction. *Information Sciences, 408*, 84-99.

Appendix

Variable	*Description*	*Variable*	*Meaning*
Age	Numeric variable that states the age of the employee	*Monthly Income*	Numeric variable indicating the monthly income of the employee
Business Travel	Nominal variable indicating whether an employee travels for work.	*Monthly Rate*	Numeric variable indicating the monthly income of the employee
Daily Rate	Numeric variable indicating the daily salary level	*Number Companies Worked*	Numeric variable indicating at how many companies the employee has worked
Department	Nominal variable indicating the department where the employee works.	*Over 18*	Nominal variable indicating whether an employee is over 18
Distance from Home	Numeric variable indicating the distance from work to home in miles.	*Over Time*	Nominal variable indicating whether an employee works overtime
Education	Ordinal variable that indicates the level of education of the employee.	*Percent Salary Hike*	Numeric variable that indicates the percentage increase in salary
Education Field	Nominal variable indicating the field of education.	*Performance Rating*	Ordinal variable indicating the level of performance of the employee
Employee Count	Numeric variable counting the number of employees	*Relationship Satisfaction*	Ordinal variable indicating the level of relationship satisfaction of the employee with the company
Employee number	Numeric variable indicating the employee ID	*Standard Hours*	Numeric variable indicating the standard hours of the employee
Environment Satisfaction	Ordinal variable indicating the level of satisfaction with the environment.	*Stock Option Level*	Ordinal variable indicating the level of stock options
Gender	Nominal variable indicating the gender of the employee.	*Total Working Years*	Numeric variable indicating the total years an employee has worked so far
Hourly Rate	Numeric variable indicating the hourly salary of the employee	*Training Times Last Year*	Numeric variable indicating the number of times an employee has had training in the past year

Job Involvement	Ordinal variable indicating the level of job involvement.	*Work Life Balance*	Ordinal variable indicating the rating an employee gives to the work life balance
Job Level	Ordinal variable indicating the level of the job.	*Years at Company*	Numeric variable indicating the number of years an employee works for the company
Job Role	Nominal variable indicating what role the employee has in the firm.	*Years in Current Role*	Numeric variable indicating the number of years the employee is in its current role
Job Satisfaction	Ordinal variable indicating the overall job satisfaction of the employee	*Years since Last Promotion*	Numeric variable indicating the number of years since the last promotion
Marital Status	Nominal variable indicating whether an employee is married.	*Years with Current Manager*	Numeric variable indicating the number of years that the employee works with the current manager

Table A.1 - *Input features*

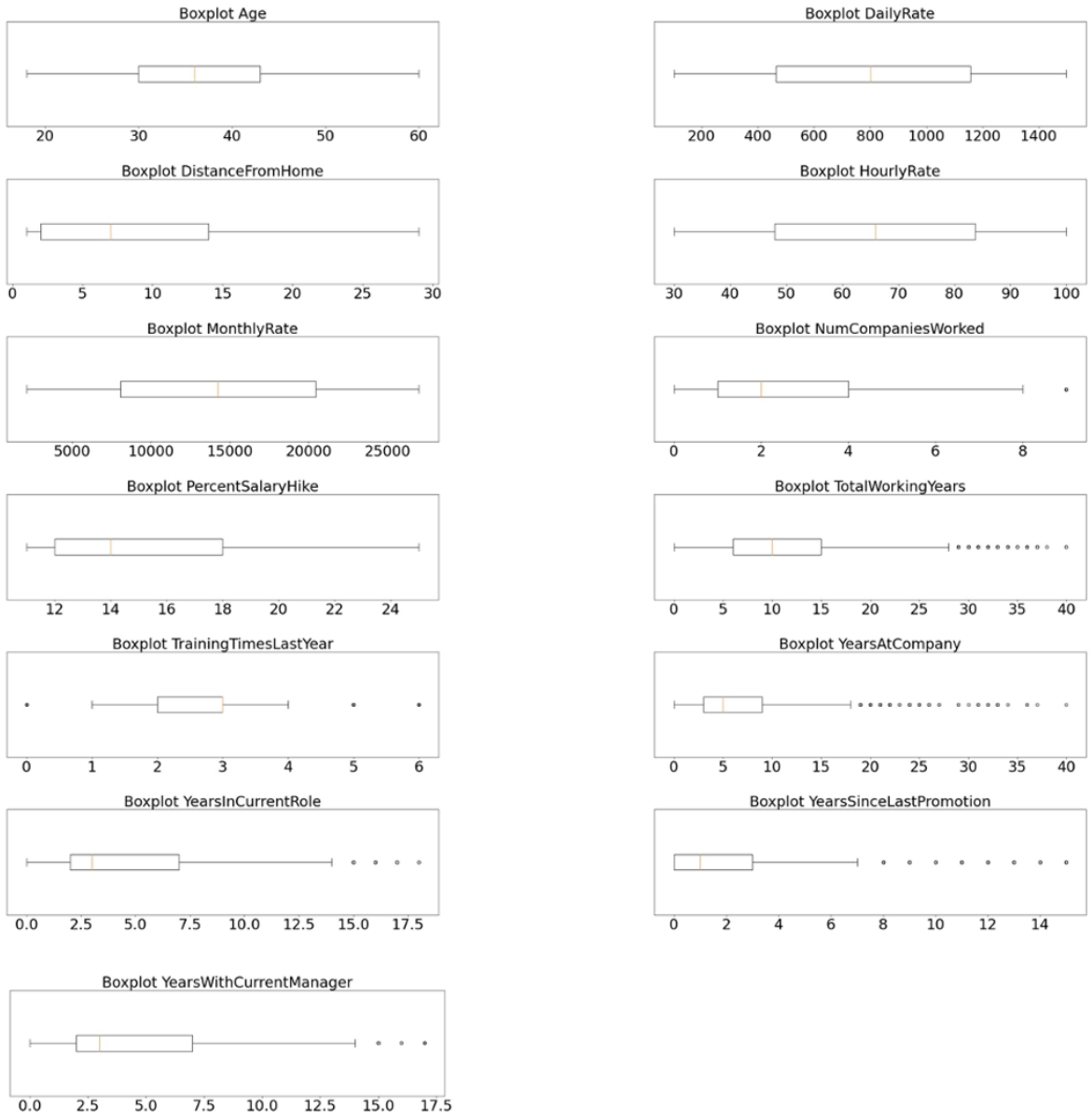

Figure A.1- *Boxplots of the variables with outliers.*

Link to Python script:
https://github.com/daniqueirisheuten/ThesisDataScience/blob/main/Thesis_programming_Danique_Heuten.ipynb