University of Wisconsin-Milwaukee
Sheldon B. Lubar School of Business

Assignment for Course:	BUS ADM 812 Machine-Learning
Submitted to:	Cheng Chen
Submitted by:	Daniel Kinkel Silva Gebhardt
Date of Submission:	12/23/2021
Title of Assignment:	Professional-grade report

CERTIFICATION OF AUTHORSHIP: I certify that I am the author of this paper and that any assistance I received in its preparation is fully acknowledged and disclosed in the paper. I have also cited any sources from which I used data, ideas of words, whether quoted directly or paraphrased. I also certify that this paper was prepared by me specifically for this course.

Student Signature:	Daniel Kinkel ***************************	Silva Gebhardt **************************

Instructor's Grade on Assignment:
Instructor's Comments:

Professional-grade Report

1. Introduction & Goals

The present paper includes a professional-grade report on a machine-learning project that fundamentally examined employee attrition. Employee attrition is the percentage of workers who leave an organization and have to be replaced by new employees. A high rate of attrition in an organization leads to increased recruitment, hiring and training costs. Some studies predict that every time a business replaces a salaried employee, it costs 6 to 9 months' salary on average. For a manager making $60,000 a year, that equates to $30,000-$45,000 in recruiting and training expenses. According to certain sources, a trillion dollars is what U.S. businesses are losing every year due to voluntary turnover.[1] Not only is it costly, but qualified and competent replacements are hard to find. In most industries, the top 20% of people produce about 50% of the output.[2] Furthermore leaving a job can bring many disadvantages on an individual level, such as the loss of income, loss of health benefits etc. A high rate of attrition in an organization can have multiple reasons for example poor management, a lack of recognition, no opportunity for growth, etc. Due to the fact that it is a multidimensional problem with a lot of drawbacks for companies and individuals, it would be of great benefit to have precise predictions of which factors cause an employee to leave the company. Companies could use these results and predictions to implement preventive measures such as optimizing recruitment, improving work conditions or creating a pleasant work culture. To get first insights on the feasibility of this activity, this paper uses HR data and different machine learning models to predict the occurrence of this phenomena on an individual level.

The goal of this project is not to produce generalizable results, but rather to get a sense of how feasible it is to predict this phenomenon using various classification models from the field of machine learning and evaluate what prediction qualities we can achieve.

2. Dataset

The dataset that we use for this purpose was published on Kaggle under the name "IBM HR Analytics Employee Attrition & Performance". It is a fictional dataset created by IBM data scientists and has been very popular with over 800k views and close to 90k downloads. The provided data consists of 1470 observations for which 34 features are given additional to the attrition information. Among those potential predictors are demographic characteristics (age, gender, education, monthly income), job characteristics (department, job role, overtime, travel frequency) as well as survey data on the overall job satisfaction and environment satisfaction. Of those 1470 fictional employees 237 are labeled to have left the company while 1233 did stay. Since the number of true values is much smaller than the number of false values, this is an imbalanced dataset, which has implications for the methodology we will apply. Running a correlation analysis shows that a variety of columns are correlated with the phenomena of Attrition. The highest one can be observed for the dummy variable specifying if the employee works overtime (0.246). Others include 'Marital Status' = 'Single' (0.175), 'Total Working Years' (-0.171) and 'Monthly Income' (-0.160). While these correlations are not particularly high, we still assume a well-trained machine learning model should be able to find patterns here that have predictive power for the attrition problem at hand.

[1] https://www.gallup.com/workplace/247391/fixable-problem-costs-businesses-trillion.aspx

[2] https://books.google.es/books?id=rx3ssrOcCTkC&lpg=RA1-PA9&ots=VaW9kZK5kV&dq=the%20top%2020%25%20of%20people%20produce%20about%2050%25%20of%20the%20output%20augustine&pg=RA1-PA11#v=onepage&q&f=false

3. Preprocessing:

Before we can start the modeling process, we have to set up the data in a proper way which is called preprossing. Since a lot of models do not have an inbuilt way to deal with those, we first check for any missing values. As there are not any included in our dataset we do not need to impute or exclude those.
Next, we apply One-hot-encoding on the nominal features in the dataset. This means that features like ‘Department’ with discrete characteristics as ‘Sales’, ‘Research & Development’ and ‘Human Resources’ are translated to dummy variables that indicate the affiliation to each of these departments.
Most importantly we split the data into a training and a validation set with a ratio of 70 to 30. This is necessary in order to obtain a valid performance measurement of the different models. Since the goal is to predict the resignations for new cases (out-of-sample forecasts), we need to withhold a part of the data from the training process. This part contains novel observations that can be used subsequently to simulate and evaluate the prediction process under realistic circumstances.
As this has been shown to have a positive effect on the performance of some ML models, we also apply Min-Max-Scaling on the data. This normalization technique changes the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values.

4. Modelling:

The models we will apply on the preprocessed training data set are K-Nearest-Neighbors, Random Forests, AdaBoosting, Support Vector Machine and Logistic Regression.
K-nearest-neighbors: The KNN model calculates the distances between an observation and all the other observations in the dataset, selects the specified number of cases(k) closest to the query, and then votes for the most frequent label (in the case of classification).
Random Forests: A Random Forest is an ensemble method that fits a set of decision tree classifiers to different, bootstrapped subsets of the dataset. The decision trees are further decorrelated by randomly selecting a subset of features for the splitting process. Subsequently the predictions of the different decision trees are averaged.
AdaBoosting: The AdaBoost algorithm is a boosting technique, which is another form of ensemble methods. It is based on weak learners like shallow decision trees that produce predictions slightly better than random guessing. When fitting the second decision tree, higher weight is given to those observations that were misclassified in by the first one. By fitting more and more learners the performance of the model is improved iteratively.
Support Vector Machine: SVM relies on a technique called kernel trick to transform the data in a higher dimensional space to find a hyperplane that best devidies the observation into the two classes. The best fit is assumed to be the one with the highest margin, which is the distance from the closest points to the hyperplane.
Logistic Regression: Logistic regression is an adaptation of the linear regression for the binary dependent variables. It predicts the probability of the event using the log function and classifies the data based on a threshold value.

Each one of them has one or more parameters that can be specified in the training process. In order to determine the best so called hyperparameters we decide on multiple candidate values for each parameter and test out all possible combinations which is called grid search.

To evaluate the performance of each parameter set we apply 3-fold cross validation. This means that the training data is randomly divided into 3 parts. Then one of the parts (folds) is picked as the test set and the other 2 serve as the training data. The model is trained on the training data and evaluated on the test data by the F1 metric that is commonly used for imbalanced datasets and will be discussed further in chapter 4. This process is repeated with the other two folds serving as the test set and the F1 scores are averaged.
After tuning the hyperparameters this way, we apply the final models on the validation set and evaluate the predictive performance in more detail.

5. Model evaluation:

Since we discuss a classification problem, we compute the confusion matrices for each model. This matrix compares the actual target values to the values predicted by the machine learning model. It creates a holistic view of how well the performance from the classification model is and what errors it is making. The confusion matrices for our models are shown in Figure X:

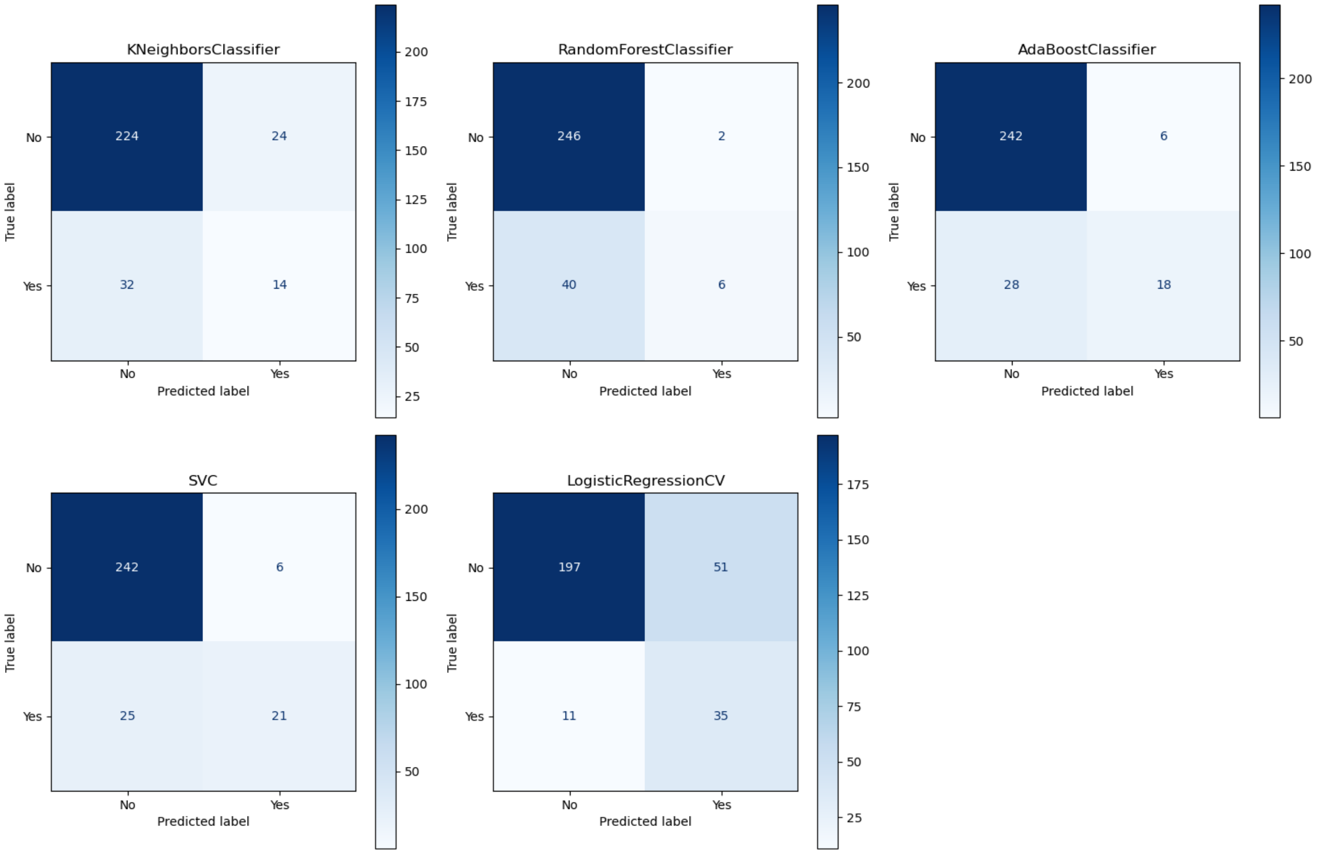

Figure 1: Confusion matrices of prediction models

Based on these values multiple metrics can be computed. A common tool to visualize the performance is the Receiver Operating Characteristic (ROC). The ROC Curve is a graph that shows the performance of a classification model at different classification thresholds. This curve represents two parameters: True Positive Rate and False Positive Rate. This can be seen as the trade-off between Sensitivity (TPR) and specificity (1 - FPR). As a baseline, the performance of a random classifier would correspond to the points along a diagonal (FPR = TPR). The further the ROC curves deviate from this baseline the better the predictive performance, which can be measured by the area under curve (AUC). One way to interpret AUC is the probability that the model will rank a random positive example higher than a random negative example. In the table below, it can be seen that Support Vector Machine with an AUC of 0.85 scores the best, followed by AdaBoosting (0.84) and Logistic Regression (0.83). The model Random Forests (0.77) and the K-Nearest-Neighbors (0.60) perform the worst. The fact that the ROC curve of KNN has such a linear shape, might be caused by the model relying on only the first nearest neighbor in the voting process.

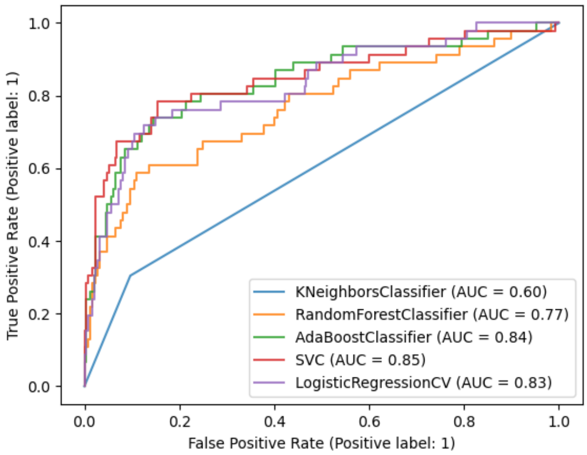

Figure 2: ROC-curves

However, since our dataset is imbalanced this measure could potentially be inaccurate. Consequently, we also compute the F1 score which is based on two other common measures, which are precision and recall. Precision indicates how many of the positive predicted cases actually turned out to be positive, and recall indicates how many of the actual positive cases that were correctly predicted by the model. The F1 score is a harmonic mean of precision and recall and thus gives a combined idea of these two metrics. By construction a model with no predictive power would yield a score of zero, while a perfect model would score one. In the table below, it can be seen that Support Vector Machine has the highest score with 0.58, followed by Logistic Regression (0.53) and AdaBoosting (0.51). The model Random Forests (0.22) and the K-Nearest-Neighbors (0.33) perform significantly worse.

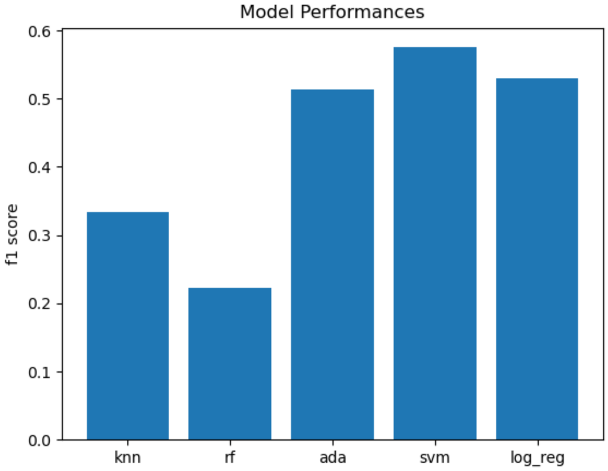

Figure 3: F1-Scores

Another important criteria for the application of machine learning models besides the performance is the interpretability of the results. In order to convince decision makers to rely on a predictive model it is important to be able to explain how the model arrives at its predictions. One aspect that is particularly helpful here is the feature importance. Feature importance refers to techniques that assign a score to input features based on how useful they are at predicting a target variable. Since our best performing model (SVM) relies on a non-linear kernel it is not possible to calculate a simple score here, we focus on AdaBoost and Logistic Regression.

The feature importance of AdaBoost is derived from the feature importance provided by the base classifier. Assuming that you use a decision tree as the base classifier, AdaBoost feature importance is determined by the average feature importance provided by each decision tree. It takes advantage of the fact that features found at the top of the tree contribute to the final prediction decision of a larger proportion of the input samples, and this expected proportion can therefore be used to estimate the relative importance of a feature. Figure X shows the ten most important input variables. ‘Monthly income’, ‘Daily Rate’, ‘Monthly Rate’ and ‘Total Working Years’ seem to be the most important features for the prediction of the model.

In order to obtain a feature importance score for the logistic regression we take the regression coefficients and scale them with the standard deviation of the feature. Below the 10 most important factors are displayed in Figure X. The years since last promotion, number of companies worked and years at the company are calculated as the most important factors.

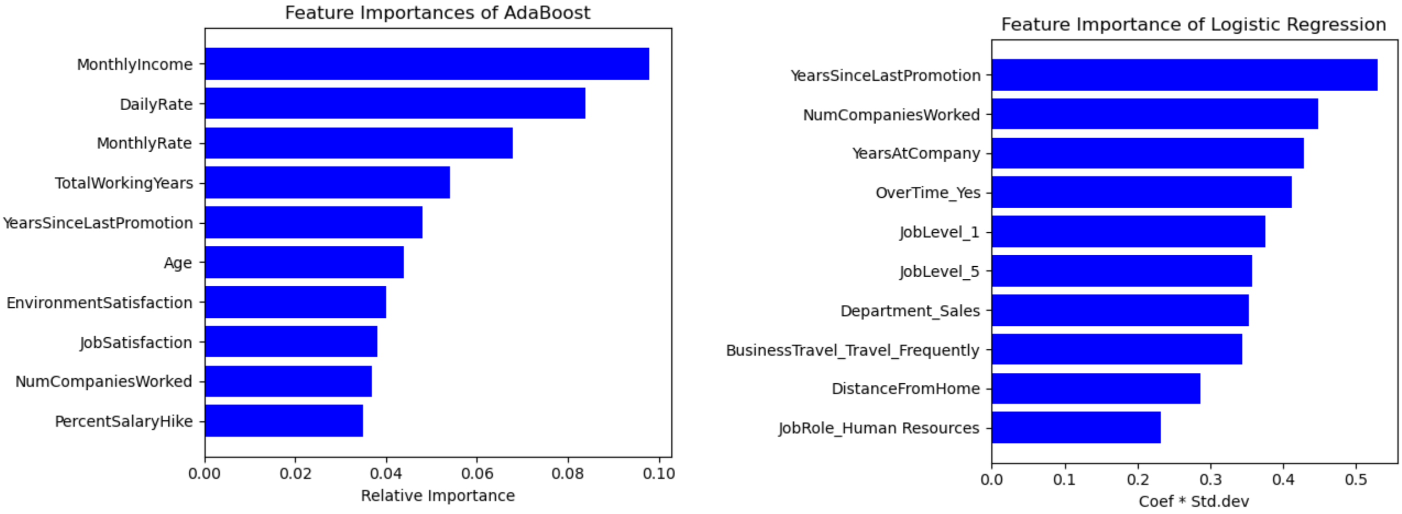

Figure 4: Feature Importances

5. Implications

After looking at the results of the different models we consider their usability in the context of our business case. Three of the compared models performed solidly (AdaBoosting, Support Vector Machine and Logistic Regression). The performance of the other two models (K-Nearest-Neighbors, Random Forests) was clearly inferior. The poor performance of random forest is especially surprising since the method is usually considered to be one of the best performing machine learning algorithms. It is possible that the preprocessing steps taken have had a negative effect on the performance of this model.

Looking at the confusion matrix of the three best performing models it is apparent that a conclusion simply based on the correctly predicted attritions cannot be made. If one compares the number of correctly predicted resignations, it seems as if the Logistic Regression-method is superior to the other ones. However, it also can also be seen that it has falsely predicted 51 observations to be positive, while AdaBoost and SVM only produced 6 of those alpha errors. However, AdaBoost and SVM did miss more cases of Attrition by falsely labeling them as negative. This is a clear case of a dilemma between type I and type II errors: for any given sample set, the effort to reduce one type of error generally results in increasing the other type of error. In the given-case of company departures, the question is, how to effectively deal with this trade-off between the alpha-error and the beta-error. Is it more harmful for the company to unnecessarily sink money into assumed employee retention and preventions, predicated on false negative predictions? Or is it less favorable for a company to contain a major alpha-error, missing positives, not inventing preventive measures and therefore losing these employees.

Unfortunately, it is not feasible to carry out a one-by-one comparison of the costs of turn-over to the total nationwide investments in prevention measures against high turnover rates, since these numbers are not possible to be determined. Therefore, it can be assumed that this trade off does not have a general optimal solution, but it remains a case-by-case decision depending on the company.

An additional factor that must be taken into account is that while SVM does slightly dominate AdaBoost when it comes to performance, AdaBoost is much better interpretable. If models are to be applied in practice, this is a very important aspect as it increases the acceptance of the prediction results. What also was surprising in this context is that AdaBoost and Logistic Regression showed very different important features. While the Adaboost model highly ranks variables that include monetary compensation, the logistic regression mostly prioritizes features relating to the job history. As this could be used as a basis for the selection of preventive measures, it would be helpful to do additional research which measures are more effective to avoid employee attrition.

6. Conclusion and evaluation of the process

Our main goal was to achieve a thorough understanding of the prediction process of a binary classification and its challenges by applying the different methodologies used in the lecture on a data set as well as analyzing and interpreting the results.

An important learning from this project is that even when large amounts of data are available, it can be difficult to come to very accurate predictions.

It also became clear that pre-processing has a very strong impact on the performance of the models, and that the sklearn module provides many helpful tools forsetting up a uniform modeling purpose and for visualizing the results.

As a highlight of the learning process, it can be noted that some of the models we trained should have high enough accuracy to help prevent attrition in this (fictitious) company.

A method that did not work as well as expected is random forest as discussed in the fifth chapter. It can be concluded that this project contributed greatly to our individual learning success by deepening the concepts of the course and therefore was an important complement of the learning process of the course.