VRIJE UNIVERSITEIT AMSTERDAM

RESEARCH PAPER

Assessment scores as a predictor of your future performance and potential

---

Author:
Joël GASTELAARS

Supervisor:
Prof. Dr. Sandjai BHULAI

A paper submitted in fulfillment of the requirements for the degree of Master of Science

in

Business Analytics
Faculty of Science

April 25, 2018

VRIJE UNIVERSITEIT AMSTERDAM

Abstract

Faculty of Science

Master of Science

Assessment scores as a predictor of your future performance and potential

by Joël GASTELAARS

Organizations these days use all kinds of assessments in their selection process to find suitable candidates for an open position. These companies believe that hiring applicants with scores above a certain threshold and with specific behavioral competencies will result in higher performance and potential scores and therefore add more economical value to the company.

This research tests the hypothesis that assessment scores can predict your future performance and potential using Linear Regression, k-Nearest Neighbor and Support Vector Machines. The predictions are validated by Root Mean Square Error and Mean Absolute Error and compared against a 'standard-3 prediction' which predicts average performance and potential scores.
The optimal models do not significantly differ from the standard prediction and there is therefore no reason to believe that the assessment variables of the provided dataset can be used to predict your future performance or potential score.

Contents

Chapter 1

Introduction

Since the 1990s, most large organizations use some sort of assessment process besides the resume screening to select suitable applicants for an open position. These assessments can be IQ-related like numerical and logical tests or more EQ-related like questionnaires about personality and behavioral competencies. These companies, supported by their research, believe that selecting candidates with scores above a certain threshold or whose behavioral traits fit within the team and matches the job profile capabilities will result in higher future job performance and potential. Research shows both supporting and contradicting evidence for the use of assessments as predictors of performance scores. As most research has been conducted in the field of Psychology, it is interesting to see whether we can give additional insights into this area with the available data.

The goal of this research is therefore simple: *"Predicting future performance and potential based on assessment scores"*. We would like to answer the research question: *"Can your assessment scores predict your future performance and potential?"*. This research uses different techniques to predict the performance and potential scores: Linear Modeling (LM), k-Nearest Neighbor (kNN) and Support Vector Machines (SVM). These different regression and classification models will be individually explained in Chapter 4.

First, Chapter 2 reviews the related work in the area of performance prediction, mostly using behavioral assessments. Chapter 3 discusses the data used for this research, beginning with data acquisition, processing the dataset and assumptions made before modeling. This includes the initial data analysis which shows the distribution of some of the variables. Chapter 4 explains the tools and techniques used and gives the theoretical background, with formulas, assumptions and error estimations of each method. It also shows the tests used to confirm these assumptions and the validation methods to check the results against. These results will be discussed in Chapter 5. Here the predictions are compared to a standard prediction. In the final chapter, Chapter 6 the significance of the results will be discussed, the implication on the business and the research questions will be answered.

Chapter 2

Literature Review

In every organization it is of high importance to make sure the employees perform well and reach their potential, as this adds economical value to the company. This goes all the way back to the selection process, where companies use different techniques to select suitable applicants with an expected high performance and potential. Several scientific articles have been published about the prediction of employee performance, most of which are in the field of Psychology. In this chapter we discuss some of the related work that has been published on this topic.

First, we have to discuss bad hires and their associated costs. The search for high performers, and therefore their characteristics, is greatly influenced by the costs of hiring and firing a low performer, e.g., a bad hire.
Boushey and Glynn (2012) took thirty case studies from eleven scientific papers on the costs of employee turnover and demonstrated that, according to this data, the median cost of turnover was 21.4% of an employee's annual salary.1 According to the U.S. Department of Labor in 1996 the associated costs of bad hires, if discovered within the first six months, were even higher and may be up to 30% of an employees first years salary.2
Especially the last statement is cited in many newspapers, articles and HR blogs on the internet. These costs can go up exponentially if the bad hire stays longer within the organization and you also take into account compensation costs and indirect costs like disruption costs and missed business opportunities according to the Society of Human Resource Management, Undercover Recruiter and others.3,4,5,6

In 1998, Schmidt and Hunter quantified that high performing employees create on average 80% more economical value for the organization than low performing employees, assuming normally distributed performance scores.7,8 Schmidt and Hunter (1998) analyzed data collected in 85 years of psychological research which resulted in three considered reliable (with validity greater than 0.5) selection methods: work samples, General Mental Ability Tests (GMAT) and structured interviews. Two years ago, Hunter et al. (2016) analyzed 100 years of data and came up with combined methods of GMAT and integrity test and GMAT and structured interview, both with a mean validity greater than 0.75.9

Richardson and Norgate (2015) state however that "considerable caution needs to be exercised in citing such correlations for test validation purposes". This is because correlations of 0.5 and higher are used to justify of the use of these assessments in selection processes while assumptions and many corrections in the data is needed to get these results. The quality of the original data is therefore also examined in their research.10

Most related work has been conducted in the field of Psychology and was a meta-analysis of the 'Big Five' personality dimensions (Extraversion, Emotional Stability, Agreeableness, Conscientiousness and Openness to Experience) and their influence on employees' performance.11,12,13,14
These five dimensions have been greatly researched since the 1960s and their influence on job performance and how to use them in the employee selection process started to have impact in the 1990s.11 The research of Barrick and Mount (1991) showed consistent relations in all researched occupations of Conscientiousness with their three job performance indicators: job proficiency, training proficiency and personnel data. For the other personality dimensions the correlations varied but the impact was small with $\rho < 0.10$. With low correlations like this it has to be seen whether it is actually possible to predict a significant difference in performance for people with higher personality traits in the Conscientiousness area than others.
Hurtz and Donovan (2000) state that much data used in the previous meta-analyses was not derived from studies that used Big Five measures. These were later categorized in the Big Five categories, which is a potential threat to the validity of the research as the data it is based on may not be classified to the Big Five personality dimensions correctly. Their research showed a similar validity and correlation for Conscientiousness as Barrick and Mount (1991) and notes that Barrick and Mount (1995) and Salgado (1997) appear to have overestimated the validity of the Big Five personality dimensions and their impact on job performance.12
Zhao and Seibert (2006) studied the differences Of the Big Five personality dimensions between entrepreneurs and managers and concluded that entrepreneurs score higher on Conscientiousness and Openness to Experience, where they score lower on Neuroticism and Agreeableness.13

According to McKenna (2002), management competencies associated with high performance can be identified. However, it is too simplistic to think that they can be represented as generalized behavioral characteristics as there are "no competencies that are truly general, but only competencies that are context-specific".15 This means that in order to get an accurate performance prediction, the required competencies should be adapted according to the job-specific context, the job personality profile.
SuccessFinder and *Technically Compatible* are two companies who got into this niche of personality assessments and claim to provide a more accurate prediction of job

performance with their assessments which cover different behavioral traits, competencies, career paths and 1000s of questions.[16,17] Larry Cash, founder of *SuccessFinder* states that personality tests have almost no validity and predict little to none of the job performance and is only useful to understand the personality and their fit in the company. They claim however, that with their method they have an accuracy in predicting job performance of 85%.[6,18] This may be true if they were able to use related behavioral traits to each specific job description. A salesperson for example obviously needs to be extravert and a secretary has to be precise and pay attention to detail. The business psychologist at *Technically Compatible* mentions that according to *SHL/CEB*, a known international assessment bureau, cheating rarely happens and most assessments have 'lie-scales' built-in. Also, follow-up interviews can be used to test some of the behavioral traits.[19]

Chapter 3

Data Analysis

3.1 Data acquisition

The datasets were provided by an international organization which prefers to stay anonymous. The data is pseudonymous to preserve data confidentiality and the privacy of the employees involved. Pseudonymous data is data where information that could be used to identify a specific person is replaced by a unique identifier, only known by the providing organization. Only that company has the unique identifiers matched to the individuals the data belongs to. The difference with anonymous data is that it is reversible. 20
The following four datasets were provided in Excel format:

- Performance data
- CAS assessment data
- LAS assessment data
- VIT assessment data

The performance data consists of the following data between 2012-2017: unique employee identifier, year, evaluation stage, evaluation score description and potential score description. All other datasets also contain this unique employee identifier, which is used to link the datasets together. The other variables in the assessment dataset are descriptive information about the employee, e.g., job-scale, region, year hired and for every category an assessment score, e.g., self-awareness, innovative, drive. It is unclear when the assessments have taken place, but it is assumed that all assessments were prerequisites to get hired. All variable names and the range of their data can be found in Appendix A.

3.2 Data processing

The performance dataset can contain multiple rows per employee, as each employee may or may not have had multiple evaluations depending on their tenure at the company. Each of the assessment datasets consists of only one row per employee. However, not all employees did all of the assessments. Only a few employees did the LAS assessments, which therefore will not be considered as a predictor of performance or potential. The analysis starts with the employees who did the CAS and

VIT assessments and have a performance and potential score. As Table 3.1 shows, these have a substantial amount of data. This is our initial merged dataset.

Dataset	**# rows**	**# unique rows**	**# unique with performance**
CAS	951	876	400
VIT	777	755	330
LAS	88	88	38
Performance	2485	612	532
Initial dataset	330	330	330

TABLE 3.1: Amount of (unique) data per dataset

To be able to use this data to train our model, the data has to be merged, cleaned and some assumptions have to be made. To merge the datasets we start with the performance dataset and add all CAS and VIT assessment data, linked using the employee's unique identifier. We start with the first row of each employee if they have multiple performance scores, as this is their oldest performance score (and therefore the first score for a new employee) and remove their other scores. The employees without any performance or potential scores are removed from the dataset. After removing unusable variables as entity, current job scale and also gender and age (as this data would also be available without conducting an assessment), the merged dataset contains 89 variables of 330 unique employees.

After merging all data in the new data-frame, it is time to have a good look at the data, clean it and make assumptions where needed. First, the data is imported into R, our preferred programming language. There the description of the performance and potential scores are removed and the potential scores are re-ordered, as we assume that a promotion within 1 year is better (and therefore should have a higher score) than a promotion in 1-3 years. All missing values are set to NA and the performance and potential scores are transformed to numerical values to calculate the correlations in Section 3.2.3.

3.2.1 Performance dataset

As mentioned in Table 3.1, the performance dataset consists of 2,485 observations of 612 unique employees between 2012-2017. Of these 612 employees, there are 80 without any performance or potential score. These will therefore be excluded from the dataset. Of the remaining 532 employees there are 23 without potential score, these will only be included as predictors for the performance score. After merging the datasets, we are left with 400 employees with performance scores and CAS assessments of which 330 also have done the VIT assessment. Other than the unique employee identifier, the performance and potential score and all assessment scores all other descriptive variables like gender and age are removed from the merged dataset as they could be derived without assessments and therefore cannot be used

as predictors in our model.

Performance Score Description	**Score**	**Potential Score Description**
Unknown	0	Unknown
Inadequate	1	Not discussed
Partially met	2	No promotion
Good	3	Growth opportunities current level
Above Expectations	4	Short-term promotion (<1 yr)
Excellent	5	Long-term promotion (1-3 yr)

TABLE 3.2: Performance & Potential Score Description

Evaluation Score Description describes on scale 0-5 how well an employee is performing and Potential Score Description also has a scale from 0-5, both described in Table 3.2. We assume that a prediction of an '0. Unknown' or '1. Not discussed' score is useless and therefore remove these score and replace them, if possible, with newer scores.

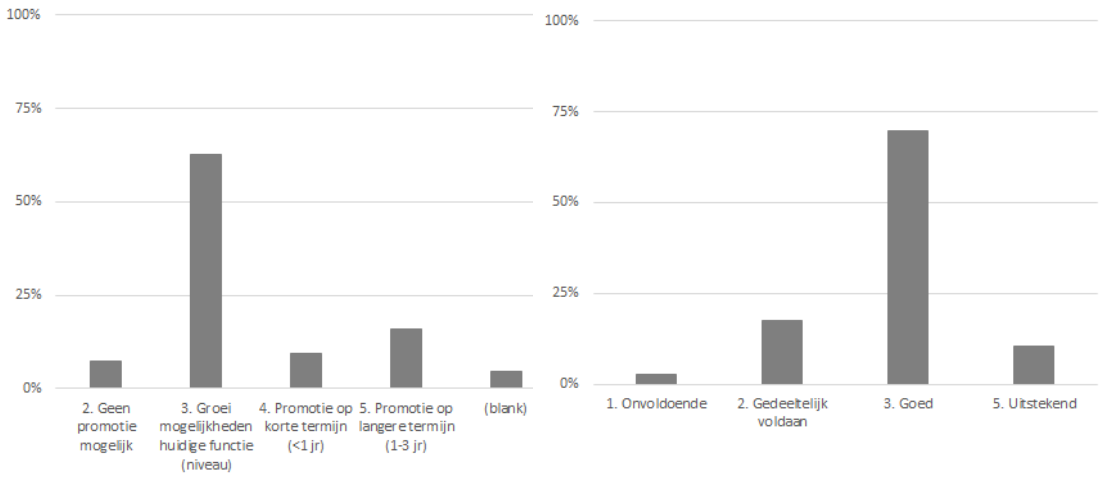

FIGURE 3.1: Potential and performance score distributions

The distributions of the cleaned performance and potential scores are plotted in Figure 3.1. Both have between 60-70% average scores (respectively 3. Good and 3. Growth opportunities current level). Where the performance scores are already in increasing order from 1-5, we decided to switch 4 and 5 of the potential scores as we assume that promotion within 1 year is better than promotion in 1-3 years.

3.2.2 Assessment datasets

All the scores of the variables in the CAS, LAS and VIT assessments are between 1-9 without missing values. The VIT assessment measures the more IQ-related variables, like numerical, logical and verbal skills. The CAS assessment is focused on the behavioral traits like drive and creativity. Two of their variable distributions are

plotted in Figure 3.2.

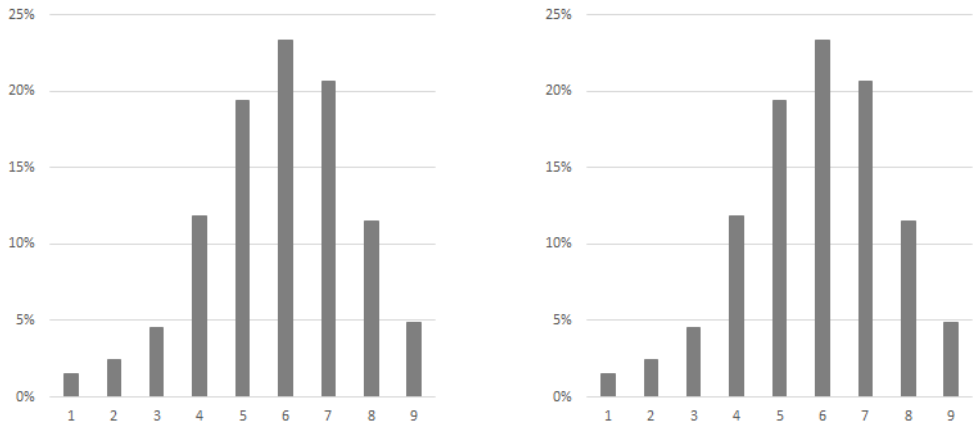

FIGURE 3.2: VIT Verbal skills score distribution (left) and CAS drive score distribution (right)

Some other variables in the assessment datasets are hiring year, gender and age. As this information is already available without applicants doing an assessment this is not taken into account in this research as a predictor for future performance and potential. The hiring year of the provided data is between 1986 and 2017 with a mean in 2013. Of the hired people with performance and CAS/VIT assessment scores 61.8% is female and the hiring age is between 19 and 59 with an average of 29.3 years. The age distribution is plotted on the left in Figure 3.3. It shows that over 76% of the employees were between 24 and 33 when hired. The educational background of employees is plotted on the right in Figure 3.3. This shows that 69% of the employees have a higher vocational education (hbo) and 10% has a university background.

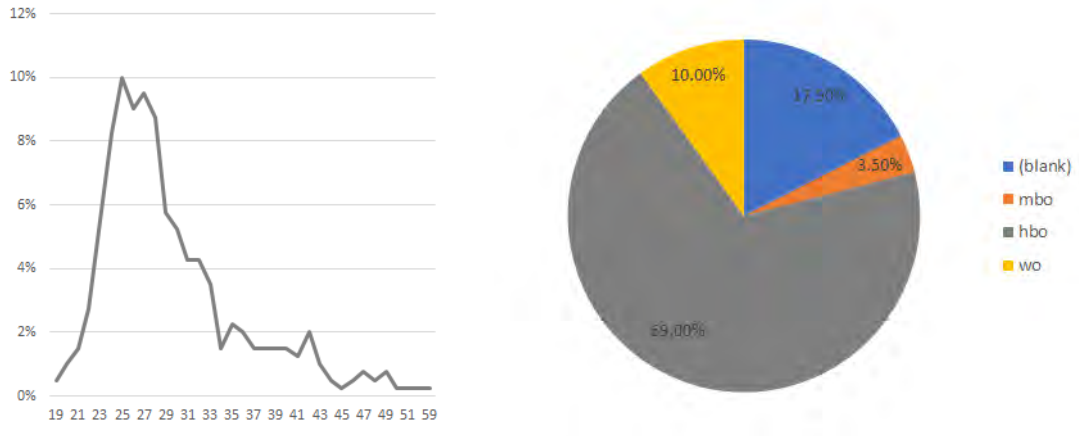

FIGURE 3.3: Hiring age distribution (left) and educational background of employees (right)

3.2.3 Correlations

Another reason not to take into account the educational background is that when calculating correlations between all variables and performance or potential scores, the educational background seems to have the highest impact, with correlations of +0.22 for university and -0.19 for 'hbo'. Of the assessment variables however, there are none with a correlation over +0.10 and only 12 (out of 86) that have a correlation lower than -0.10 with the performance score. These are:

- Interest
- Sensation seeking
- Imagination
- Culture for change
- Involvement
- Listening skills
- Service-orientated
- Willingness to change
- Analysis and judgment
- Creativity
- Innovation
- Environmental awareness

There are two positively correlated assessment variables with values higher than +0.10 with the potential score. These are 'Performance Motivation' and 'Negotiation'. The other individual correlations of all variables with the performance and potential scores can be found in Appendix A. As the correlations found are very low, the expectation is that it will be tough to conclude that assessments are a significant predictor for future performance or potential.

3.2.4 Assumptions

1. All assessments were prerequisites to get hired and are therefore potential predictors for the first performance and potential scores of applicants.
2. Only assessment data which is available before hiring is a potential predictor of performance. And all data which could be received without conducting assessments is ignored, as these would also be available without assessments.
3. In case of multiple scores, we assume that the oldest available performance and potential score are the first given scores and therefore the ones to predict.
4. In case the oldest performance score does not have a potential score, the next potential score is used if there is any and vice versa.
5. A short-term promotion (within 1 year) is 'better' than a long-term promotion (within 1-3 years).
6. Unknown or not discussed scores are irrelevant.

Chapter 4

Methodology

4.1 Linear Regression

Linear regression models the relationship between the response variable y_i and explanatory variables $x_{i,1}$ to $x_{i,j}$, where j is the amount of explanatory variables used. The goal of linear regression is to plot a line through the data points and minimize the distance of the points to the line. For this dataset we use a multivariate ordinal regression several explanatory variables:

$$y_i = \beta_0 + \beta_1 * X_{i,1} + \ldots + \beta_j * X_{i,j} + \epsilon_i \tag{4.1}$$

Where in this case i = 1, 2 and j = 1, 2, ..., k. And $k \leq 86$. Where response variables y_1 and y_2 are the performance score and potential score, explanatory variable $X_{i,j}$ refers to assessment score j and ϵ_i measures the error of response variable i. There are j+1 coefficients β which are estimated and show the effect of each explanatory variable on the response variable.

4.1.1 Assumptions

The following assumptions have to be tested to be able to correctly interpret the coefficients and therefore use the linear model in the right way. If an assumption fails to be true, this does not mean linear regression cannot be used. This only means that the estimator is not necessarily the maximum likelihood estimator and the results can be unreliable.

1. **Independence of errors.** The residuals ϵ_i are independently distributed and there is no correlation between the errors.

   To check the independence of errors for time series, the autocorrelation function (acf) can be plotted or the Durbin-Watson test can be used. Because this is not a time series model, it is enough to show that the residuals are randomly distributed and therefore do not correlate with each other. The Runs test can be used to check the randomness of the residuals' distribution. This will be shortly explained in Section 4.1.2.

2. **Normality of errors.** The residuals are normally distributed with expected value $\mathbb{E}(\epsilon_i) = 0$ and variance σ^2: $\epsilon_i \sim N(0, \sigma^2)$.

   The residuals should be normally distributed around 0, otherwise the results could be unreliable. There are several ways to check the normality of the residuals e.g., using a qq-plot, the (Robust) Jarque-Bera test, D'Agostino-Pearson test, Shapiro-Wilk test or Kolmogorov-Smirnov test. The used tests in this research are shortly explained in Section 4.1.2.

3. **Homoscedasticity.** All errors are assumed to have approximately the same variance for different values of the response variables and are therefore uncorrelated with the explanatory variables: $var(\epsilon_i|X_i) = \sigma_\epsilon^2$ for $i = 1, \ldots, n$.

   If the homoscedasticity of the residuals does not hold, than the significance tests of the estimated coefficients β would be unreliable. This is called heteroscedasticity. However, it does not necessarily mean that the estimators are biased.

4. **Weak exogeneity.** The explanatory variables X are assumed to be free of (measurement) errors.

5. **Linearity.** The relationship between the response variables and explanatory variables is assumed to be linear, as demonstrated in equation 4.1.

   The linear relationship between the response variables and explanatory variables can be tested by plotting the data, although more complex relationships may be hard to find. To check possible non-linear relations, the variables can be added to the model and checked for their significance.

6. **Lack of perfect (multi)collinearity.** The correlation of any two of the used explanatory variables X is not perfect e.g., unequal to 1 or -1.

   If this does not hold, it could influence the estimated coefficients β as its variance would increase.

4.1.2 Used tests

This section will shortly explain all tests used to check the assumptions of Linear Regression.

Runs test

The Runs test is used to check the randomness of the distribution of the residuals. Therefore is has the null hypothesis H_0: the residuals are randomly distributed. H_0 is rejected if the resulting p-value is lower then the significance level α.

The test statistic is formulated as:

$$Z = \frac{R - \overline{R}}{\sigma_R}$$
$$\text{With} \quad \overline{R} = \frac{2n_1n_2}{n_1 + n_2} + 1 \tag{4.2}$$
$$\text{And} \quad \sigma_R^2 = \frac{2n_1n_2(2n_1n_2 - n_1 - n_2)}{n_1 + n_2(n_1 + n_2 - 1)}$$

Where n_1 and n_2 are the number of positive (above median) and negative (below median) values. Its visualization looks like Figure 4.1, where the red A's stand for the positive values and the blue B's for the negative values.

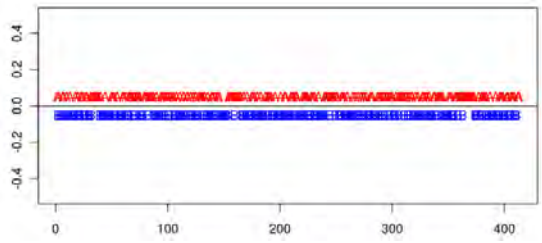

FIGURE 4.1: Runs test visualization

(Robust) Jarcque-Bera test

The (Robust) Jarcque-Bera (JB) test checks whether the residuals have the skewness and kurtosis of a normal distribution. It has therefore the null hypothesis H_0: The residuals are normally distributed. H_0 is rejected if the resulting p-value is lower then the significance level α.
The test statistic of JB can be mathematically written as follows:

$$JB = \frac{(n - k + 1)}{6}(S^2 + \frac{1}{4}(C - 3)^2) \tag{4.3}$$

Where n is the number of observations/degrees of freedom, S the skewness of the data, C the kurtosis of the data and k the amount of explanatory variables/number of regressors.

$$S = \frac{\hat{\mu_3}}{\hat{\sigma}^3} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2)^{3/2}} \tag{4.4}$$

$$C = \frac{\hat{\mu_4}}{\hat{\sigma}4} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2)^2} \tag{4.5}$$

Where $\hat{\mu_3}$ is the estimation of the third moment, $\hat{\mu_4}$ the estimation of the fourth moment, $\bar{x}$ the sample mean and σ^2 the variance.

QQ-plot

The QQ-plot, R function *qqnorm* from the package *stats*, can be used as a visual check of the normality assumption of the residuals. The residual values are plotted against the normal distribution, a line in the form of $y = ax + b$ with $a, b \in \mathbb{R}^+$. In case of a normal distribution, the residual values will be approximately around the line, as in Figure 4.2.

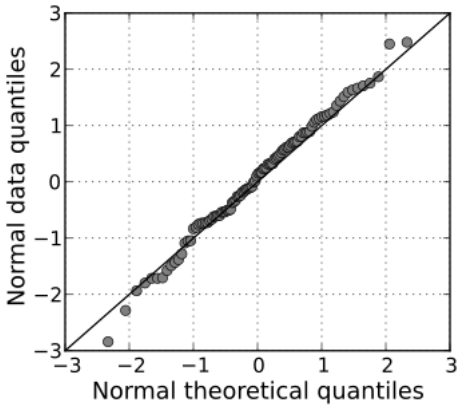

FIGURE 4.2: QQ-plot of normal distributed data

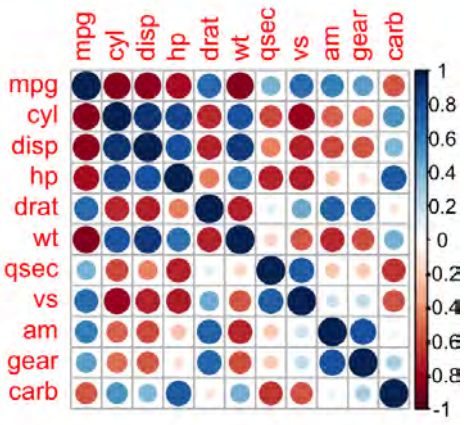

FIGURE 4.3: Correlation plot example

Correlation plot

To check the lack of perfect multicollinearity the R function *corrplot* from the package *corrplot* provides a visualization of the correlations between all variables. This can be used to see whether there are still explanatory variables with high correlation in the model, where darkblue is high positive correlation and darkred is high negative correlation, see Figure 4.3.

4.2 K-Nearest Neighbor

K-Nearest Neighbor (kNN) is a non-parametric, lazy machine learning algorithm which takes the k most similar neighbors using a similarity measure to classify an object (classification) or provide it with a value (regression). As kNN is a non-parametric and lazy algorithm, there are no assumptions made about the data distribution and there is a minimal training phase, all training data is used in the testing phase. The only assumption is that the data is in a metric space, which means that there is a way to measure similarity (e.g., distance) between variables. A commonly used way to calculate the similarity is the Euclidean distance. The number k refers to the amount of neighbors that affect the classification or regression.[24,25]

We can use both kNN classification and regression in this research, they will be explained in the subsections below.

4.2.1 Classification

Figure 4.4 gives an example of how kNN classification works. The green circle is the new observation, ready to be classified as a blue square (class 1) or a red triangle (class 2). First, the similarity between the new observation and the training instances are calculated. Then, the classifications of the k training instances with the highest similarity (in case of Euclidean distance therefore the lowest distance) are checked and the new example is classified to the category of the majority. In this case, for k=1 it is assigned to the first class and for k=3 it is assigned to the second class. When an even number of k is chosen and there is a 'tie', a random decision will be made depending on the settings of your model.

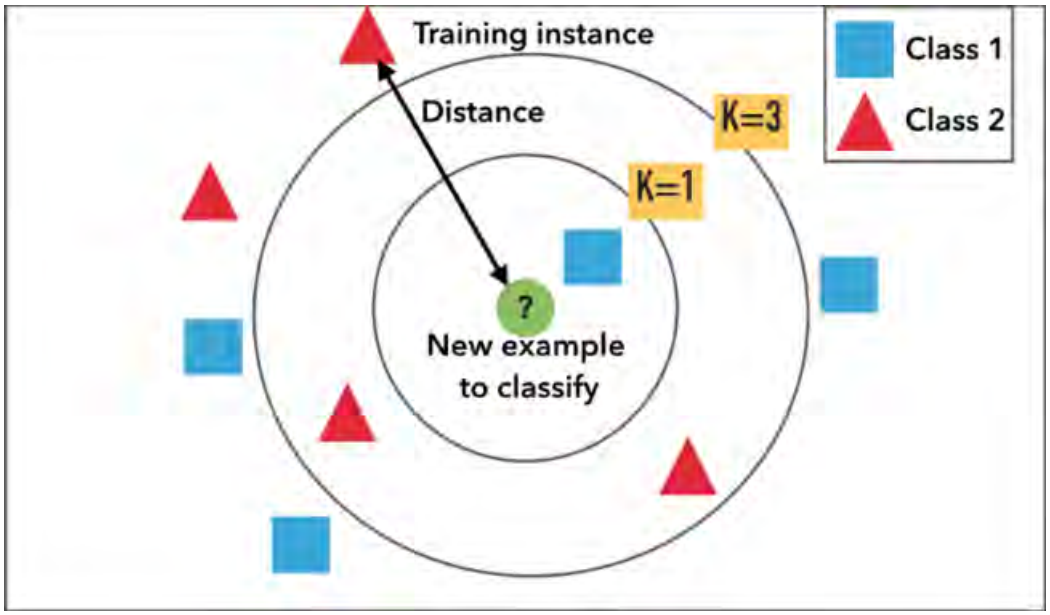

FIGURE 4.4: kNN classification example. 26

In this research there are four classes for performance (1,2,3,5) and also four classes for potential (2,3,4,5) which makes the classification a bit harder but it works similar as the example. Therefore both classification and regression models are used. For classification, the R function *knn* from the package *class* is used, which calculates the Euclidean distance when provided with the training set, test set, number of neighbors k considered and factor of true classifications of the training set. After this its classifies the object according to the majority vote. 27

4.2.2 Regression

In the kNN regression model the performance or potential scores of the k most similar employees are averaged to provide the new object with an averaged performance/potential score:

$$y_{new} = \frac{\sum_{i=1}^{k} y_i}{k} \tag{4.6}$$

Where y_{new} is the performance or potential score of the new employee and y_i the score of of the i-th most similar employee, according to the Euclidean distance. The R function *knn.reg* from the *FNN* package is used, which calculates the Euclidean distance when provided with the training set, test set, number of neighbors k considered and factor of true values of the training set. After this the average of its k neighbors is assigned to the object. 28

4.3 Support Vector Machine

Just like k-Nearest Neighbor, the Support Vector Machine algorithm (SVM) is a non-parametric technique which does not make any assumptions about the data. The basic concept of Support Vector Machines is to design a hyperplane to divide all instances into two sets. The best hyperplane is the one which maximizes the margin $\gamma = 2/||w||$ of the two closest points of the classes, which are called the support vectors.
Figure 4.5 intuitively shows how this works. The optimal hyperplane separates the two target groups with the maximum margin. The six points on the margin hyperplanes are now the support vectors for this model.

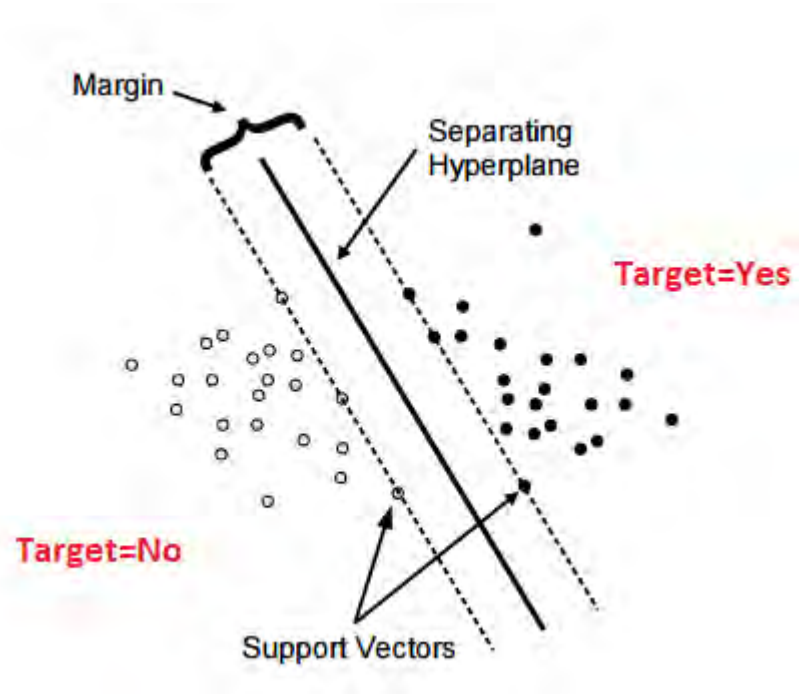

FIGURE 4.5: SVM classification example. 29

SVMs use kernels to linearly separate the data points and reduces the influence of misclassified instances by adding cost C per misclassification, also called the 'soft margin hyperplane'. Maximizing the margin gives us a unique global minimum,

another advantage of SVMs. Mathematically, this can also be formulated as the following minimizing quadratic optimization problem:

$$\begin{aligned} \min_{w,b} \quad & Q(w) = \frac{1}{2}||w||^2 + C\sum_{i}^{\ell}\xi_i \\ \text{w.r.t.} \quad & y_i(w \cdot x_i + b) \geq 1 - \xi_i, \quad \forall x_i, \quad i = 1, \ldots, \ell. \end{aligned} \tag{4.7}$$

With $y_i \in$ Class {-1,1} and $\xi_i \geq 0$. 30,31,32,33,35,36,37,38.

This means that a SVM can basically only solve classifications with two classes. To solve multi-class classification problems, these are split up and solved by the 'one-against-one' or 'one-against-all' techniques. The 'one-against-one' approach basically splits the multi-class problem into binary class problems for each pair of classes and solves these accordingly. When combined, a voting scheme is applied to all K(K-1)/2 sub-problems and the classifier with the most '+1' predictions gets assigned by the combined classifier. The 'one-against-all' technique builds a SVM for each class, separating it from all other classes. When combined, the class labels of the classifiers with the highest confidence score are assigned.

4.3.1 The *e1071* package

In this research the function *svm* from the R package *e1071* is used for both the classification and regression models of the SVM calculations. The performance of SVMs strongly depends on the parameters given to the model: cost C (of the regularization term in the Lagrange formulation), kernel type, regression or classification type and the kernel parameters. These parameters can be tuned manually or with the *tune* function to find the best fit for the model.

The *e1071* package has a R interface with the awarding winning C++ *libsvm* library from Chang and Lin (2001). When there are k>2 classes to classify, the *libsvm* library uses the 'one-against-one' approach by training all subset classifiers and using a voting system to provide the right class. A sparse data representation is used which saves computational time, as only non-zero values are stored.

4.3.2 The *svm* function

As mentioned before, the performance of a SVM strongly depends on the parameters given to the model. The *svm* function has the several input parameters to vary or choose from.

Formula. Just like for linear regression the formula represents the model to be fitted.
Data. This contains the test set sample data.

Class.weights. This can be used to provide asymmetric class sizes different weights.
Cost. Costs of violation of the constraints.
Epsilon (ϵ). Used to control the width of the ϵ-insensitive zone to fit the training data. The bigger ϵ the fewer support vectors.
Type. As SVMs can be used for both classification and regression, there are choices to be made:

- C-classification
- nu(ν)-classification
- one-classification (for novelty detection)
- eps(ϵ)-regression
- nu(ν)-regression

Nu (ν). A parameter for nu-classification and nu-regression.
Kernel. There are four possible kernels: Linear, Polynomial, Radial and Sigmoid. Each of which has its own parameters.
Gamma (γ). A parameter for all but linear kernels.
Coef0. A parameter for polynomial and sigmoid kernels.

There is also a **cross** input parameter possible, which is used for k-fold cross validation of the training data to assess the model's quality.

4.3.3 The *tune* function

The function *tune* helps finding the optimal parameters for the Support Vector Machine. It uses randomized 10-fold cross validation to split the training set and tunes with performance measure the classification error or the mean squared error. The first one is used for classification SVMs and the latter for regressions SVMs.

4.4 Validation

There are several ways to validate how accurate the used models are in predicting performance and potential score in this research. First of all, a standard prediction is made by prediction that all future employees have an average performance and potential score, which in both case is a 3. Then, the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are calculated on the test set for the standard prediction and all used methods.

Cross-validation is also a commonly used validation method, which is already built in the packages *FNN* (10-fold cross validation using function *knn.cv*) and *e1071* (k-fold cross validation using function *svm* with parameter **cross**=k). In this research,

we use this on the training set, creating a validation set to optimize the parameters and prevent overfitting.

4.4.1 Mean Absolute Error

The Mean Absolute Error (MAE) measures the absolute differences between the prediction and the actual values, the size of the average prediction error. In mathematical form, we can write this as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.8}$$

With y_i the actual value and $\hat{y}_i$ the predicted value of i.

4.4.2 Root Mean Square Error

The Root Mean Square Error (RMSE) also measures the difference between the predicted values and actual values. It represents the sample standard deviation of the residuals.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.9}$$

With y_i the actual value and $\hat{y}_i$ the predicted value of i.

The RMSE gives a relatively higher value to large errors. RMSE is therefore more desirable to use than MAE when large errors are to be penalized (e.g., a misclassification by 2 instead of 1 is twice as bad).

Chapter 5

Results

In this research three different techniques were tested and compared against a 'standard-3 prediction', which meant predicting an average performance and potential score for every new employee.
After merging, cleaning and preparing the data as explained in Chapter 3, the data had to be split in a training set and test set. We used the rule of thumb to split this in an 80% training set and 20% test set, as this left enough instances to test the models on.

5.1 Linear Regression

To check the assumptions made for linear regression, we did several (visual) tests, as explained in Chapter 4. First, to test the independence of the residuals, we did the Runs test from the *lawstat* package. The result was a p-value of 0.267 which is not enough to reject H_0: the residuals are randomly distributed. In Figure 5.1 there is no clear pattern to be found.

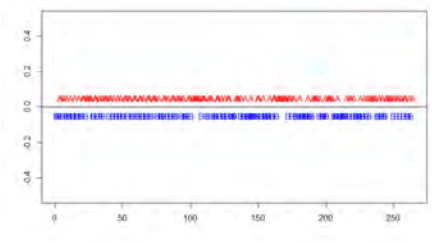

FIGURE 5.1: Runs test visualization

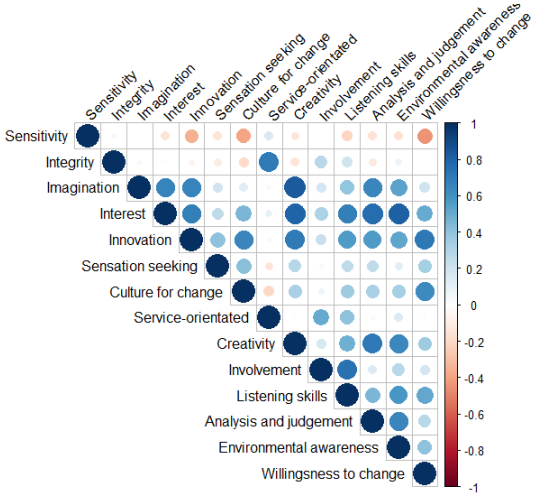

FIGURE 5.2: Correlation plot of 14 possible predictors

After creating a correlation matrix of all 86 assessment variables, some variables with minimal correlation between themselves were selected for the linear regression. This to prevent multicollinearity, which was also visually tested with the potential

explanatory variables in Figure 5.2.

The mean of the residuals is 3.5e-17, which is almost 0. However, the Jarcque-Bera test got us a p-value of 6.7e-16 which means they are not normally distributed. This can also been seen in the qq-plot at the top-right in Figure 5.3. This means that the linear regression estimator could be unreliable.

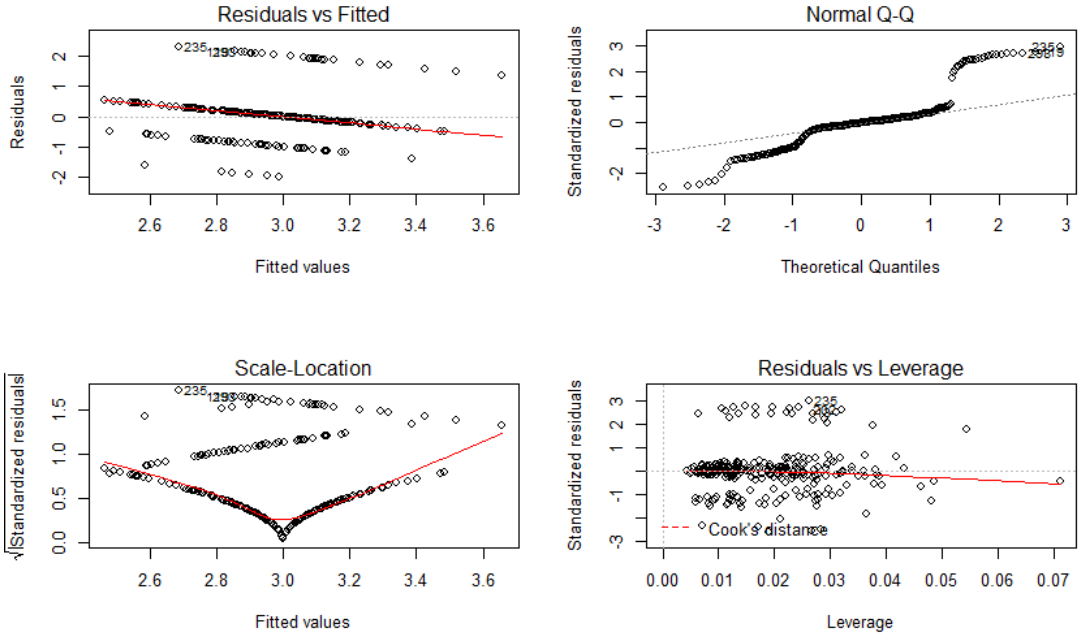

FIGURE 5.3: Several plots of the residuals in the linear regression model

The top-left and bottom-left plots in Figure 5.3 show that there is still a bit heteroscedasticity, as there is a slightly downward trend in the top-left figure. However, when tested with the *gvlma* package the homoscedasticity assumption is acceptable. For all different models the variables used were uncorrelated with the residuals, using Pearson's product of moment correlation which makes assumption about homoscedasticity acceptable.

Several different combinations of independent variables were tested, and the two best models are compared in Table 5.1. The model with the lowest Root Mean Square

	Performance		**Potential**	
Variables	**RMSE**	**MAE**	**RMSE**	**MAE**
1,3,6,8	0.767	0.455	0.578	0.431
1,3,6	**0.763**	**0.438**	**0.578**	**0.430**

TABLE 5.1: Linear Regression results

Error (RMSE) and Mean Absolute Error (MAE) was model with assessment variables 1,3,6: Sensitivity, Imagination and Sensation Seeking. The other model also included the variable Service-oriented and provided similar results, depending on the sample. Mathematically the best models can be written as:

$$\begin{aligned} y_1 &= 3.26 + 0.03X_{1,1} - 0.06X_{1,2} - 0.01X_{1,3} \\ y_2 &= 3.54 + 0.03X_{1,1} - 0.02X_{1,2} - 0.05X_{1,3} \end{aligned} \tag{5.1}$$

5.2 k-Nearest Neighbor

As mentioned in Chapter 4, both the classification and regression methods of k-Nearest Neighbor (kNN) were used in this research. For both methods, the amount of neighbors k used was varied between 1 and 50 to find the optimal k with the lowest RMSE and MAE. To put the results into perspective, they were plotted against the RMSE and MAE of the 'standard-3 prediction'. In Figure 5.4 the two plots on the top show the kNN classification method and its RMSE for performance (top-left) and potential (top-right), where the bottom two plots show the kNN regression method and its RMSE for performance (bottom-left) and potential (bottom-right).

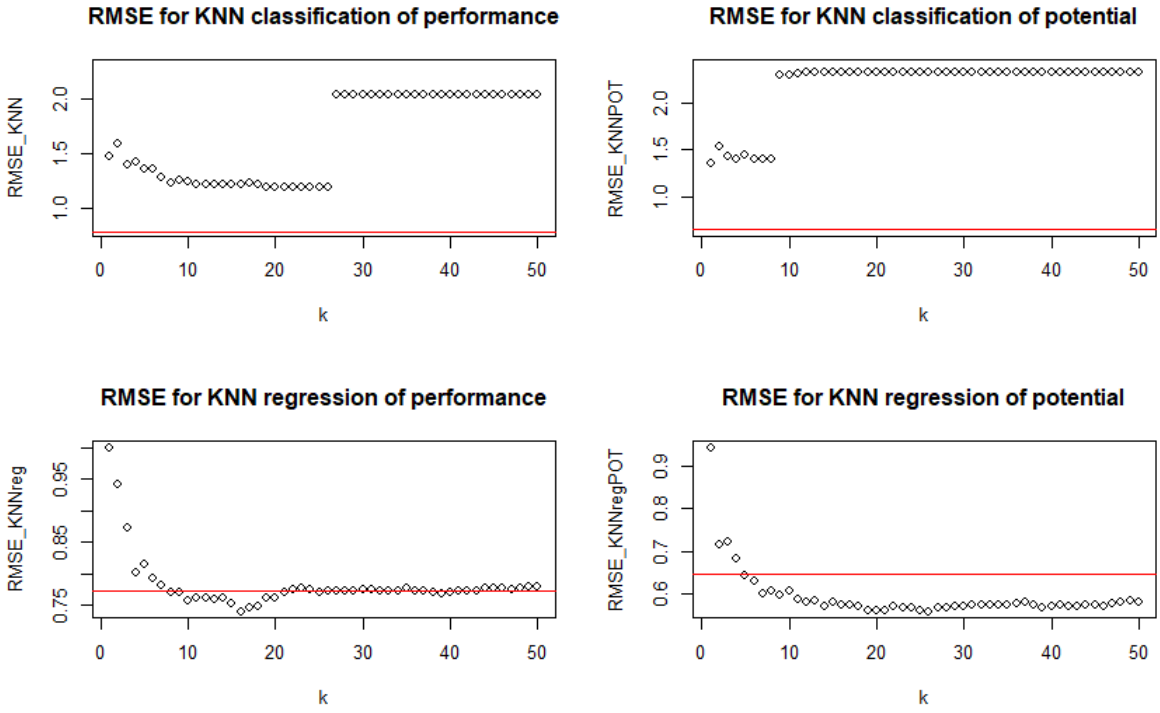

FIGURE 5.4: kNN classification (top) and regression (bottom) RMSE for the prediction of performance (left) and potential (right) score.

Figure 5.4 showed that the kNN regression model has for both performance and potential a lower RMSE than the classification model. It also showed that the RMSE

is lower than the 'standard-3 prediction' and that around 15 and 25 neighbors seems to get the lowest RMSE for respectively performance and potential for this sample. The MAE however, is higher than the MAE of the standard-3 prediction, which is shown in Table 5.3.

5.3 Support Vector Machine

For the Support Vector Machine algorithm (SVM) one classification and two regression methods were compared. The nu(ν)-classification method was infeasible as it has hard bounds on allowed misclassification which could not be satisfied.

	Performance			**Potential**		
Method	**Cost**	**Gamma**	ϵ/ν	**Cost**	**Gamma**	ϵ/ν
C-classification	10	0.5	NA	1	2	NA
Nu-regression	0.1	2	0.5	1	2	0.5
Eps-regression	0.1	0.5	0.1	1	2	0.1

TABLE 5.2: Optimal parameters for SVM models

After tuning the models using 10-fold cross validation on the training set, the optimal parameters of all SVMs used a radial kernel and 251 support vectors. Table 5.2 shows the other optimal parameters used for the three different models.

5.4 Model errors

In Table 5.3 all model errors are compared and the method with the lowest RMSE and MAE are highlighted for both the performance and potential score predictions.

	Performance		**Potential**	
Method	**RMSE**	**MAE**	**RMSE**	**MAE**
Standard-3 prediction	0.773	**0.371**	0.648	**0.290**
Linear Regression (optimal model)	0.763	0.438	0.578	0.430
k-Nearest Neighbor (classification)	1.191	1.000	1.362	1.178
k-Nearest Neighbor (regression)	**0.740**	0.421	**0.559**	0.424
Support Vector Machine (C-classification)	0.773	**0.371**	0.596	0.466
Support Vector Machine (eps-regression)	0.764	0.428	0.595	0.455
Support Vector Machine (nu-regression)	0.771	0.377	0.773	0.466

TABLE 5.3: Scores

Chapter 6

Discussion & Conclusion

The aim of this research was to find a model in which assessment scores can be used as a predictor for future performance and potential scores. Three different techniques were used to test the hypothesis: *Assessment scores can be used as a predictor for your future performance and potential scores*. The results in Chapter 5 however, show that these predictions are basically the same as predicting an average performance (3. Good) and potential score (3. Growth opportunities current level) for all new employees.

Linear Regression

The optimal linear regression model does not meet the normality assumption of the residuals. However, the residuals are randomly distributed with mean 0 and no correlation with each other or the explanatory variables. The homoscedasticity assumption does not seem to hold according to the model plot, but the R function *gvlma* says it is still an acceptable assumption. All in all, it seems reasonable to say that the estimator of the linear regression is possibly unreliable. The predictions of performance between 2.7 and 3.3 and potential between 3.0 and 3.7 are mainly because of β_0, as equation 5.1 revealed.

k-Nearest Neighbour

The results for kNN show only for the KNN regression models a slightly lower RMSE than the standard prediction. And again, the prediction lie between respectively 2.8 and 3.2 for the performance scores and 3.1 and 3.6 for the potential scores. In Figure 5.4 you can see that a k>10 seems to get lower errors. However, this only means that the KNN regression method is converging to the standardized value. The bigger k gets, the more values it averages and the closer the range of values predicted around 3.

Support Vector Machines

The results for SVM show minimal improvements over the standard prediction. And again the predicted values lie around 3. Tuning the versions of SVM, did not make much difference. Neither did weighing the classes, which is unexpected as there are

significantly more '3' values then other values and putting a higher cost on misclassifying the other values could have changed things.

Limitations and future research

It is important to note that the provided datasets with assessment and performance scores are relatively small compared to the datasets large organizations might have. As some research mentioned, splitting the data into several job categories might help finding 'suitable' behavioral traits for that job. Still, especially because most companies already use a certain threshold which reduces the spread, it is not certain that with bigger datasets any proclaimed patterns will be found.

Impact and summary

Even though some researchers in Psychology claim to have proven that some behavioral traits can predict future performance, nothing in this research supports their claim. Little correlation is found between any of the variables and the performance or potential score. The 'standard-3 prediction' has comparable RMSE and MAE as the optimized models, which makes them irrelevant. With these results there is no reason to believe that your assessment scores can be used as predictor of your future performance or potential.

Bibliography

[1] Boushey, H., Glynn, S.J. (2012). *There Are Significant Business Costs to Replacing Employees*. Center for American Progress.

[2] Hacker, C.A. (1996). *The cost of bad hiring decisions & how to avoid them*. Cornell University, NY.

[3] Frye, L. (2017). *The Cost of a Bad Hire can be Astronomical*. Society for Human Resource Management. https://www.shrm.org/resourcesandtools/hr-topics/employee-relations/pages/cost-of-bad-hires.aspx.

[4] Sunberg, J. (2017). *What is the true cost of hiring a bad employee?*. Undercover Recruiter. https://theundercoverrecruiter.com/infographic-what-cost-hiring-wrong-employee/.

[5] Deering, S. (2017). *The cost of bad hires and how to avoid them*. Undercover Recruiter. https://theundercoverrecruiter.com/cost-bad-hire-avoid/.

[6] Cash, L. (2017). *Predicting Job Performance: Do Personality Tests Work?*. https://www.successfinder.com/predicting-job-performance-do-personality-tests-work/.

[7] Cannate, D. (2014). *Don't use the crystal ball: Top selection methods to predict employees performance*. https://scienceforwork.com/blog/dont-use-the-crystal-ball-top-selection-methods-to-predict-employees-performance/.

[8] Schmidt, F.L. and Hunter, J.E. (1998). *The Validity and Utility of Selection Methods in Personnel Psychology: Practical and Theoretical Implications of 85 Years of Research Findings*. Psychological Bulletin, Vol 124, No. 2, pp. 262-274.

[9] Schmidt, F.L., Oh, I. and Shaffer, J.A. (2016). *The Validity and Utility of Selection Methods in Personnel Psychology: Practical and Theoretical Implications of 100 Years of Research Findings*. Fox School of Business Research Paper. Available at SSRN: https://ssrn.com/abstract=2853669.

[10] Richardson, K. and Norgate, S.H. (2015). *Does IQ Really Predict Job Performance?*. Applied Development Science, 19:3, pp. 159-169, https://doi.org/10.1080/10888691.2014.983635.

[11] Barrick, M.R. and Mount, M.K. (1991). *The big five personality dimension and job performance: a meta-analysis*. Department of Management and Organizations, University of Iowa. Personnel Psychology, Vol. 44. https://doi.org/10.1111/j.1744-6570.1991.tb00688.x.

[12] Hurtz, G.M. and Donovan, J.J. (2000). *Personality and Job Performance: The Big Five Revisited*. Journal of Applied Psychology, Vol. 85, No. 6, pp. 869-879. DOI: 10.1037//0021-9010.85.6.869.

[13] Zhao, H. and Seibert, S.E. (2006). *The Big Five Personality Dimensions and Entrepreneurial Status: A Meta-Analytical Review*. Journal of Applied Psychology, Vol. 91, No. 2, pp. 259-271. DOI: 10.1037/0021-9010.91.2.259.

[14] Dudley, N.M., Orvis, K.A., Lebiecki, J.E. and Cortina, J.M. (2006). *A Meta-Analytic Investigation of Conscientiousness in the Prediction of Job Performance: Examining the Intercorrelations and the Incremental Validity of Narrow Traits*. Journal of Applied Psychology, Vol. 91, No. 1, pp. 40-57. DOI: 10.1037/0021-9010.91.1.40.

[15] McKenna, S. (2002). *Can knowledge of the characteristics of "high performers" be generalised?*. Journal of Management Development, Vol. 21, No. 9, pp.680-701, https://doi.org/10.1108/02621710210441676.

[16] SuccessFinder. https://www.successfinder.com/.

[17] Technically Compatible. https://www.technicallycompatible.com.

[18] Cash, L. (2017). *Predicting Job Performance: Behavioral Assessments Work!*. https://www.successfinder.com/predicting-job-performance-behavioral-assessments-work/.

[19] Guest author (2017). *HOW PERSONALITY ASSESSMENT HELPS PREDICT PERFORMANCE*. Sten10 Business Psychologists. https://www.technicallycompatible.com/how-personality-assessment-helps-predict-performance/.

[20] Williamsen, C. (2017). *PSEUDONYMIZATION VS. ANONYMIZATION AND HOW THEY HELP WITH GDPR*. http://www.protegrity.com/pseudonymization-vs-anonymization-help-gdpr/.

[21] Guha, R., Wagner, T., Darling-Hammond, L., Taylor, T., & Curtis, D. (2018). *The promise of performance assessments: Innovations in high school learning and college admission*. Palo Alto, CA: Learning Policy Institute.

[22] Abassi, S.M., Hollman, K.W. (2000). *Turnover: The Real Bottom Line*. Public Personnel Management. Vol. 21, No. 3, pp. 333-342.

[23] Prabhakaran, S. (z.d.). *Assumptions of Linear Regression*. http://r-statistics.co/Assumptions-of-Linear-Regression.html.

[24] Thirumuruganathan, S. (2010). *A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm*. https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/.

[25] Bronhstein, A (2017). *A Quick Introduction to K-Nearest Neighbors Algorithm*. https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7.

[26] Christo, E. (2017). *Classification with KNN – Cancer Diagnosis Example*. http://mymljourney.com/classification-with-knn-and-cancer-diagnosis-example/.

[27] R Documentation. *k-Nearest Neighbour Classification*. https://stat.ethz.ch/R-manual/R-devel/library/class/html/knn.html.

[28] R Documentation. *k-Nearest Neighbour Regression*. https://www.rdocumentation.org/packages/FNN/versions/1.1/topics/knn.reg.

[29] DnI Institute (2015). http://dni-institute.in/blogs/building-predictive-model-using-svm-and-r/.

[30] R Documentation. *Support Vector Machines*. https://www.rdocumentation.org/packages/e1071/versions/1.6-8/topics/svm.

[31] Chang, C. and Lin, C (2011). *LIBSVM: a library for Support Vector Machines*. ACM Transactions on Intelligent Systems and Technology, Vol. 2, Issue: 3, pp. 27:1 – 27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[32] Fan, R. and Chen, P. and Lin, C. (2005).*Working Set Selection Using the Second Order Information for Training SVM*. Journal of Machine Learning Research, Vol. 6, pp. 1899-1918. http://www.csie.ntu.edu.tw/~cjlin/papers/quadworkset.pdf.

[33] *Support Vector Machine Regression*. http://kernelsvm.tripod.com/.

[34] The *e1071* package in R, containing the *svm* and *tune* functions. https://cran.r-project.org/web/packages/e1071/e1071.pdf.

[35] Meyer, D. (2017). *Support Vector Machines*.

[36] Premalatha, M. and Lakshmi, V. (2013). *SVM TRADE-OFF BETWEEN MAXIMIZE THE MARGIN AND MINIMIZE THE VARIABLES USED FOR REGRESSION*. International Journal of Pure and Applied Mathematics, Vol. 87, No. 6, pp. 741-750.

[37] Bennett, K.P. and Campbell, C. (2000). *Support Vector Machines: Hype or Hallelujah?*. SIGKDD Explorations, Vol. 2, Issue: 2, pp. 1-13.

[38] Cortes, C. and Vapnik, V. (1995). *Support Vector Networks*. Machine Leaming, Vol. 20, pp. 273-297.

[39] *Forget GPAs And Test Scores: Performance Assessment Can Predict Who Gets Accepted and Who Gets Hired*. https://www.forbes.com/sites/avivalegatt/2018/03/01/forget-gpas-and-test-scores-performance-assessment-can-predict-who-gets-accepted-and-who-gets-hired/#70f033526d44.

Appendix A

Data Tables

A.1 Table A.1 Variable names and range

Variable name	Range	Average
gevoeligheid	1-9	4.56
extraversie	1-9	5.99
interesse	1-9	5.55
mensgerichtheid	1-9	5.30
taakgerichtheid	1-9	5.53
g1_nervositeit	1-9	4.98
g2_boosheid	1-9	4.87
g3_neerslachtigheid	1-9	4.27
g4_gene	1-9	4.60
g5_stressgevoeligheid	1-9	4.70
e1_vriendelijkheid	1-9	5.89
e2_contactbehoefte	1-9	5.87
e3_dominantie	1-9	5.69
e4_dynamiek	1-9	5.29
e5_spanningsbehoefte	1-9	5.86
e6_opgewektheid	1-9	6.06
i1_verbeeldingskracht	1-9	5.32
i2_artistieke_interesse	1-9	4.96
i3_emotionaliteit	1-9	5.54
i4_veranderingsgezindheid	1-9	5.53
i5_intellectuele_interesse	1-9	5.32
i6_vrijzinnigheid	1-9	5.39
m1_vertrouwen	1-9	5.31
m2_integriteit	1-9	5.29
m3_betrokkenheid	1-9	5.97
m4_meegaandheid	1-9	4.74
m5_bescheidenheid	1-9	4.98
m6_compassie	1-9	5.17
t1_zelfverzekerdheid	1-9	5.58

Variable name	**Range**	**Average**
t2_ordelijkheid	1-9	4.98
t3_gewetensvol	1-9	5.55
t4_prestatiemotivatie	1-9	5.86
t5_zelfdiscipline	1-9	5.79
t6_bedachtzaamheid	1-9	5.05
Impressiemanagement	1-3	1.20
Zelfdeceptie	1-3	1.06
Accuratesse	1-9	5.26
Besluitvaardigheid	1-9	5.63
Delegeren	1-9	5.48
Kwaliteitsgerichtheid	1-9	5.36
Onderhandelen	1-9	5.36
Plannen	1-9	5.14
Plichtsbesef	1-9	5.59
Presenteren	1-9	5.67
Presteren onder druk	1-9	5.40
Resultaatgerichtheid	1-9	5.54
Structureren	1-9	5.40
Sturen	1-9	5.70
Aanpassingsvermogen	1-9	5.55
Contactvaardigheid	1-9	5.85
Draagvlak creeren	1-9	5.50
Feedback geven	1-9	5.62
Klantgerichtheid	1-9	5.96
Luistervaardigheid	1-9	5.87
Motiveren	1-9	5.84
Onderzoeken van drijfveren	1-9	5.63
Organisatiesensitiviteit	1-9	5.44
Overtuigen	1-9	5.75
Samenwerken	1-9	5.93
Teambuilding	1-9	5.80
Assertiviteit	1-9	5.48
Commerciele drive	1-9	5.84
Dienstverlenend	1-9	5.34
Drive	1-9	5.64
Dynamiek	1-9	5.36
Flexibiliteit	1-9	5.16
Initiatief	1-9	5.53
Integriteit	1-9	5.06
Ondernemerschap	1-9	5.70
Sensitiviteit	1-9	5.75

Variable name	Range	Average
Stressbestendigheid	1-9	5.29
Veranderingsbereidheid	1-9	5.67
Zelfontwikkeling	1-9	5.50
Analyseren en oordeelsvorming	1-9	5.28
Creativiteit	1-9	5.42
Helicopterview	1-9	5.46
Innoveren	1-9	5.65
Marktgerichtheid	1-9	5.73
Omgevingsbewustzijn	1-9	5.25
Strategisch inzicht	1-9	5.37
MPI_Basisaanleg	1-9	5.53
Totaalscore	1-9	6.26
Cijfermatig redeneervermogen	1-9	5.71
Logisch redeneervermogen	1-9	5.98
Rekenvaardigheid	1-9	6.34
Verbale aanleg	1-9	6.10

A.2 Table A.2 Correlations

Variable name	Correlation performance score	Correlation potential score
gevoeligheid	0.0601	0.0749
extraversie	-0.0807	-0.0549
interesse	-0.1461	-0.0590
mensgerichtheid	-0.0726	-0.1638
taakgerichtheid	0.0351	0.0142
g1_nervositeit	0.0604	0.0556
g2_boosheid	0.0044	0.0741
g3_neerslachtigheid	0.0472	0.0879
g4_gene	0.0365	0.0336
g5_stressgevoeligheid	0.0229	0.0512
e1_vriendelijkheid	-0.0755	-0.0926
e2_contactbehoefte	-0.0490	-0.0858
e3_dominantie	0.0545	0.0485
e4_dynamiek	0.0800	0.0699
e5_spanningsbehoefte	-0.1730	-0.0227
e6_opgewektheid	-0.0758	-0.0482

Variable name	**Correlation performance score**	**Correlation potential score**
i1_verbeeldingskracht	-0.1326	-0.0115
i2_artistieke_interesse	-0.0362	-0.0518
i3_emotionaliteit	-0.0358	-0.0697
i4_veranderingsgezindheid	-0.1401	0.0072
i5_intellectuele_interesse	-0.0977	-0.0744
i6_vrijzinnigheid	-0.0799	-0.0286
m1_vertrouwen	-0.0027	-0.0902
m2_integriteit	-0.0062	-0.1190
m3_betrokkenheid	-0.1022	-0.0653
m4_meegaandheid	-0.0325	-0.0881
m5_bescheidenheid	-0.0353	-0.0732
m6_compassie	-0.0681	-0.1438
t1_zelfverzekerdheid	0.0019	-0.0173
t2_ordelijkheid	0.0236	0.0447
t3_gewetensvol	0.0050	-0.0808
t4_prestatiemotivatie	0.0441	0.1014
t5_zelfdiscipline	-0.0462	-0.0791
t6_bedachtzaamheid	0.0368	0.0953
Impressiemanagement	-0.0128	-0.0098
Zelfdeceptie	0.0054	0.0881
Accuratesse	0.0214	0.0172
Besluitvaardigheid	-0.0049	-0.0395
Delegeren	0.0521	0.0443
Kwaliteitsgerichtheid	0.0619	0.0579
Onderhandelen	0.0416	0.1209
Plannen	0.0081	0.0000
Plichtsbesef	-0.0005	-0.1137
Presenteren	-0.0254	0.0286
Presteren onder druk	-0.0381	-0.0420
Resultaatgerichtheid	-0.0028	0.0451
Structureren	-0.0678	-0.0699
Sturen	0.0253	0.0752
Aanpassingsvermogen	-0.0691	-0.1192
Contactvaardigheid	-0.0435	-0.0909
Draagvlak creeren	0.0521	0.0508
Feedback geven	0.0758	0.0132
Klantgerichtheid	-0.0795	-0.1000
Luistervaardigheid	-0.1053	-0.0653
Motiveren	-0.0008	0.0031
Onderzoeken van drijfveren	-0.0642	-0.0617
Organisatiesensitiviteit	-0.0235	-0.0170

Variable name	**Correlation performance score**	**Correlation potential score**
Overtuigen	-0.0456	0.0070
Samenwerken	-0.0562	-0.1089
Teambuilding	-0.0711	-0.1325
Assertiviteit	0.0136	0.0305
Commerciele drive	-0.0673	-0.0173
Dienstverlenend	-0.1380	-0.1830
Drive	0.0539	0.0729
Dynamiek	0.0076	0.0302
Flexibiliteit	-0.0777	0.0182
Initiatief	-0.0057	-0.0386
Integriteit	-0.0236	-0.1375
Ondernemerschap	-0.0231	0.0121
Sensitiviteit	-0.0658	-0.1007
Stressbestendigheid	-0.0351	-0.0702
Veranderingsbereidheid	-0.1405	-0.0301
Zelfontwikkeling	0.0161	0.0606
Analyseren en oordeelsvorming	-0.1335	-0.0542
Creativiteit	-0.1368	-0.0425
Helicopterview	-0.0984	0.0033
Innoveren	-0.1445	-0.0192
Marktgerichtheid	-0.0875	0.0497
Omgevingsbewustzijn	-0.1001	0.0366
Strategisch inzicht	-0.0785	0.0318
MPI_Basisaanleg	0.0147	0.0278
Totaalscore	-0.0374	0.0287
Cijfermatig redeneervermogen	-0.0745	-0.0561
Logisch redeneervermogen	0.0199	0.0683
Rekenvaardigheid	-0.0458	0.0640
Verbale aanleg	-0.0304	-0.0209

Appendix B

R-code

B.1 JWG_ResearchPaperBA2018.R

```
# Research Paper Business Analytics - April 2018
# Joel Gastelaars - 2132710
#
# Supervised by: Prof. dr. Sandjai Bhulai
#
# Assessment scores as a predictor of your future performance and potential.
#

rm(list =ls())

library(dplyr)
library(data.table)
library(base)
library(ggplot2)
library(readxl)
library(stats)
library(class)
library(FNN)
library(base)
library(hydroGOF)
library(e1071)
library(lmtest)
library(lawstat)
library(gvlma)
library(corrplot)
library(RColorBrewer)

setwd("C:/Users/Joel/Documents/Business Analytics/Master/Research Paper/Research data/")

#Import data
merged <- read.csv("perfVIT-17.csv", header = TRUE) #313 rows with performance, potential, VIT and CAS data

#####Preparing dataset
#Removing comments from scores
merged$Evaluation.Score.Description <- as.numeric(substr(merged$Evaluation.Score.Description, 0, 1))
merged$Potential.Score.Description <- as.numeric(substr(merged$Potential.Score.Description, 0, 1))

#Re-arranging potential scores, as short term promotion (within 1 yr) is 'better' than long-term (in 1-3 years)
merged$Potential.Score.Description[merged$Potential.Score.Description == 4] <- 6
merged$Potential.Score.Description[merged$Potential.Score.Description == 5] <- 4
merged$Potential.Score.Description[merged$Potential.Score.Description == 6] <- 5

#Focus first on CAS/VIT with performance AND POTENTIAL scores
table(merged$Evaluation.Score.Description)

#Calculate correlations PERFORMANCE
cor_vector = NULL
for (i in 4:89) {
  cor_vector[i-3] <- cor(merged[,i],merged$Evaluation.Score.Description, use = "complete.obs", method = "pearson")
}
summary(cor_vector)
cor_vector

#Check names of all correlations with higher absolute value then 0.1
useful_pos_cor <- 3 + which(cor_vector > 0.1) #positive corr > 0.1
colnames(merged[useful_pos_cor])
useful_neg_cor <- 3 + which(cor_vector < -0.1) #negative corr <-0.1
colnames(merged[useful_neg_cor])
```

```
  #Calculate correlations POTENTIAL
  cor_vectorPOT = NULL
  for (i in 4:89) {
    cor_vectorPOT[i-3] <- cor(merged[,i],merged$Potential.Score.Description, use = "complete.obs", method = "pearson")
  }
  summary(cor_vectorPOT)
  cor_vectorPOT

  #Check names of all correlations with higher absolute value then 0.1
  useful_pos_corPOT <- 3 + which(cor_vectorPOT > 0.1) #positive corr > 0.1
  colnames(merged[useful_pos_corPOT])
  useful_neg_corPOT <- 3 + which(cor_vectorPOT < -0.1) #negative corr <-0.1
  colnames(merged[useful_neg_corPOT])

  table(merged$Potential.Score.Description)


############## Predict Performance & Potential Scores

##Create train and testset 80/20 %
index <- 1:nrow(merged)
testIndex <- sample(index, trunc(length(index)/5))
#62 instances test
testset <- merged[testIndex,]
check = testset$Evaluation.Score.Description #performance score
checkPOT = testset$Potential.Score.Description #potential score
#Remove perf/pot score and employee key
testset[1:3] <- NULL

#251 instances train
trainset = merged[-testIndex,]
cl = trainset$Evaluation.Score.Description #performance score
clPOT = trainset$Potential.Score.Description #potential score
trainset[1:3] <- NULL
#Remove perf/pot score and employee key


#########################################################################################################################

####LINEAR REGRESSION
#variance-cov matrix
tempset <- matrix(, nrow = 251, ncol = 14)
tempset[,1] = trainset$gevoeligheid
tempset[,2] = trainset$Integriteit
tempset[,3] = trainset$i1_verbeeldingskracht
tempset[,4] = trainset$interesse
tempset[,5] = trainset$Innoveren
tempset[,6] = trainset$e5_spanningsbehoefte
tempset[,7] = trainset$i4_veranderingsgezindheid
tempset[,8] = trainset$Dienstverlenend
tempset[,9] = trainset$Creativiteit
tempset[,10] = trainset$m3_betrokkenheid
tempset[,11] = trainset$Luistervaardigheid
tempset[,12] = trainset$Analyseren.en.oordeelsvorming
tempset[,13] = trainset$Omgevingsbewustzijn
tempset[,14] = trainset$Veranderingsbereidheid


#Best combinations: 1,3,6,8 (RMSE, MAE)  SOMETIMES (DIFF SAMPLES) depending on MAE or RMSE preference.

covmat = matrix(c(cov(tempset)), nrow =14, ncol =14)
#cor matrix
cormat=cov2cor(covmat)
rownames(cormat) <- c("Sensitivity","Integrity","Imagination","Interest","Innovation","Sensation seeking","Culture for change",
                      "Service-orientated","Creativity","Involvement","Listening skills","Analysis and judgement",
                      "Environmental awareness","Willingsness to change")
colnames(cormat) <- c("Sensitivity","Integrity","Imagination","Interest","Innovation","Sensation seeking","Culture for change",
                      "Service-orientated","Creativity","Involvement","Listening skills","Analysis and judgement",
                      "Environmental awareness","Willingsness to change")

#No perfect multicollinearity <<-- OK
corrplot(cormat, type = "upper", method = "circle", tl.col="black", tl.srt=45)

linear_model <- lm(cl ~ gevoeligheid + interesse + Integriteit, data=trainset)
pred2 <- predict.lm(linear_model,testset)
RMSE_LM <- rmse(pred2,check)
MAE_LM <- mae(pred2,check)

linear_modelPOT <- lm(clPOT ~ gevoeligheid + interesse + Integriteit, data=trainset)
pred2POT <- predict.lm(linear_modelPOT,testset)
RMSE_LMPOT <- rmse(pred2POT,checkPOT)
```

```
MAE_LMPOT <- mae(pred2POT,checkPOT)

#linear regression worse then "standard-3"


############################Check assumptions LM - http://r-statistics.co/Assumptions-of-Linear-Regression.html
#model equation is linear <<- NOT OK, plot(merged$Evaluation.Score.Description, merged$gevoeligheid)

#mean error is 0 <<- OK
mean(linear_model$residuals)
mean(linear_modelPOT$residuals)

#no autocorrelation of residuals, randomness residuals <<- OK
#acf(linear_model$residuals)#(time series)
lawstat::runs.test(linear_model$residuals, plot.it=TRUE) # p-value > 0.05 means H0: residuals are random, no patterns
lawstat::runs.test(linear_modelPOT$residuals, plot.it=TRUE)
#lmtest::dwtest(linear_model) #(time series) p-value > 0.05 means H0: autocorrelation is 0
## YES INDEPENDENT ERRORS (time series)
#Box.test(linear_model$residuals^2, lag=12, type="Ljung-Box")#p-value>0.05 means H0: residuals are independent distributed


## NOT NORMAL DIST ERRORS
rjb.test(linear_model$residuals, option="JB")# p-value>0.05 means H0: residuals are normal distributed
shapiro.test(linear_model$residuals) #not normal
rjb.test(linear_modelPOT$residuals, option="JB")# p-value>0.05 means H0: residuals are normal distributed
shapiro.test(linear_modelPOT$residuals) #not normal

#The X variables and residuals are uncorrelated, high p-value means no rejection h0= 0 corr. <<<- OK
cor.test(trainset$gevoeligheid, linear_model$residuals)  # check for every used variable
cor.test(trainset$Integriteit, linear_model$residuals)
cor.test(trainset$i1_verbeeldingskracht, linear_model$residuals)
cor.test(trainset$e5_spanningsbehoefte, linear_model$residuals)
cor.test(trainset$Dienstverlenend, linear_model$residuals)
#CHECK SAME FOR POTENTIAL
cor.test(trainset$gevoeligheid, linear_modelPOT$residuals)  # check for every used variable
cor.test(trainset$Integriteit, linear_modelPOT$residuals)
cor.test(trainset$i1_verbeeldingskracht, linear_modelPOT$residuals)
cor.test(trainset$e5_spanningsbehoefte, linear_modelPOT$residuals)
cor.test(trainset$Dienstverlenend, linear_modelPOT$residuals)


#Homoscedasticity <<- OK, normality of residuals <<-- NOT OK
par(mfrow=c(2,2))
plot(linear_model) #PERFORMANCE
plot(linear_modelPOT) #POTENTIAL
par(mfrow=c(1,1))

#Number of obs must be greater than number of X <<-- OK
var(trainset$gevoeligheid) #3.81
var(trainset$Integriteit) #3.
var(trainset$i1_verbeeldingskracht) #3.21
var(trainset$Dienstverlenend) #3.18
var(trainset$e5_spanningsbehoefte) #3.18


#TEST ALL <<--- 3 NOT OK
gvlma::gvlma(linear_model, alphalevel = 0.05) #PERFORMANCE
gvlma::gvlma(linear_modelPOT, alphalevel = 0.05) #POTENTIAL

#############################################################################################################################

## K Nearest Neighbours CLASSIFICATION & REGRESSION
###########PERFORMANCE
RMSE_KNN = NULL
MAE_KNN = NULL
RMSE_KNNreg = NULL
MAE_KNNreg = NULL
#RMSE_CV = NULL
#MAE_CV = NULL
for(j in 1:50) {
  knn <- knn(train = trainset,test = testset,cl,k = j) #knn classification
  knnreg <- knn.reg(train = trainset,test = testset,cl,k = j) #knn regression

#  knn_cv <- knn.cv(trainset, cl, k = j, prob = FALSE)  #Cross-validation

  RMSE_KNN[j] = rmse(as.numeric(knn),check)
  MAE_KNN[j] = mae(as.numeric(knn),check)
  RMSE_KNNreg[j] = rmse(as.numeric(knnreg$pred),check)
  MAE_KNNreg[j] = mae(as.numeric(knnreg$pred),check)
# RMSE_CV[j] = rmse(as.numeric(knn_cv),as.numeric(cl))
# MAE_CV[j] = mae(as.numeric(knn_cv),as.numeric(cl))
```

```
}
#COMPARE TO PREDICTING ALL 3S -- STANDARD PREDICTION, all average performances
standard = NULL
for(i in 1:length(knn)){
  standard[i] = 3
}
RMSE_STANDARD = rmse(standard,check)
MAE_STANDARD = mae(standard,check)


##################POTENTIAL
RMSE_KNNPOT = NULL
MAE_KNNPOT = NULL
RMSE_KNNregPOT = NULL
MAE_KNNregPOT = NULL
#RMSE_CV = NULL
#MAE_CV = NULL
for(j in 1:50) {
  knnPOT <- knn(train = trainset,test = testset,clPOT,k = j) #knn classification
  knnregPOT <- knn.reg(train = trainset,test = testset,clPOT,k = j) #knn regression

  #  knn_cv <- knn.cv(trainset, cl, k = j, prob = FALSE)  #Cross-validation

  RMSE_KNNPOT[j] = rmse(as.numeric(knnPOT),checkPOT)
  MAE_KNNPOT[j] = mae(as.numeric(knnPOT),checkPOT)
  RMSE_KNNregPOT[j] = rmse(as.numeric(knnregPOT$pred),checkPOT)
  MAE_KNNregPOT[j] = mae(as.numeric(knnregPOT$pred),checkPOT)
  # RMSE_CV[j] = rmse(as.numeric(knn_cv),as.numeric(cl))
  # MAE_CV[j] = mae(as.numeric(knn_cv),as.numeric(cl))
}


#COMPARE TO PREDICTING ALL 3S -- STANDARD PREDICTION, all average performances
standardPOT = NULL
for(i in 1:length(knnPOT)){
  standardPOT[i] = 3
}
RMSE_STANDARDPOT = rmse(standardPOT,checkPOT)
MAE_STANDARDPOT = mae(standardPOT,checkPOT)

##################### COMBI PLOT PERFORMANE AND POTENTIAL

#The KNN RMSE and MAE converge to the values of the standard prediction
par(mfrow=c(2,2))
plot(RMSE_KNN, xlab = "k", ylim = c(0.8,2.3), main = "RMSE for KNN classification of performance") + abline(h=RMSE_STANDARD, col = "red")
#plot(MAE_KNN, xlab = "k", ylim = c(0.4,2.1)) + abline(h=MAE_STANDARD, col = "red")
plot(RMSE_KNNPOT, xlab = "k", ylim = c(0.65,2.4), main = "RMSE for KNN classification of potential") + abline(h=RMSE_STANDARDPOT, col = "red")

plot(RMSE_KNNreg, xlab = "k", main = "RMSE for KNN regression of performance") + abline(h=RMSE_STANDARD, col = "red")
#plot(MAE_KNNreg, xlab = "k", ylim = c(0.4,0.8)) + abline(h=MAE_STANDARD, col = "red")
plot(RMSE_KNNregPOT, xlab = "k", main = "RMSE for KNN regression of potential") + abline(h=RMSE_STANDARDPOT, col = "red")

par(mfrow=c(1,1))

# standard = NULL
# for(i in 1:length(knn_cv)){
#   standard[i] = 3
# }
# RMSE_CVSTANDARD = rmse(standard,as.numeric(cl))
# MAE_CVSTANDARD = mae(standard,as.numeric(cl))
#
#Best solution at k=4
#plot(RMSE_CV) + abline(h=RMSE_CVSTANDARD)
#plot(MAE_CV) + abline(h=MAE_CVSTANDARD)

#When K=10 you get the 'optimal' prediction which converges with RMSE and MAE to the standard prediction: All 3's
#Therefore KNN does not give satisfied results.




#########################################################################################################################

#####SUPPORT VECTOR MACHINES ------------------------- PERFORMANCE
#nu-regression, eps-regression or c-classification
svm_model <- svm(cl ~ ., data=trainset, type = "nu-classification")
summary(svm_model)
pred <- predict(svm_model,trainset)
summary(pred)
#table(pred,cl)
```

```
svm_tune <- tune(svm, train.x=trainset, train.y=cl,
                 kernel="radial", ranges=list(cost=10^(-1:2), gamma=c(.5,1,2)))

print(svm_tune)

svm_model_after_tune <- svm(cl ~ ., data=trainset, kernel="radial", type = "C-classification", cost=10, gamma=0.5)
summary(svm_model_after_tune)
## CHECK ON TESTSET
pred <- predict(svm_model_after_tune,testset)
summary(pred)

RMSE_SVM <- rmse(as.numeric(pred),check)
MAE_SVM <- mae(as.numeric(pred),check)


#tuned SVM scores worse then "standard-3" prediction, but close.

#####SUPPORT VECTOR MACHINES ---------------------------------- POTENTIAL
#nu-regression, eps-regression or c-classification
#class.weights only for classification, but doesnt seem to affect prediction...
#cost <- table(clPOT)
#cost[2]=100000000000000000
svm_modelPOT <- svm(clPOT ~ ., type = "nu-regression", data=trainset)
summary(svm_modelPOT)
predPOT <- predict(svm_modelPOT,trainset)
summary(predPOT)
#table(pred,cl)

svm_tunePOT <- tune(svm, train.x=trainset, train.y=clPOT,
                 kernel="radial", type = "nu-regression",   ranges=list(cost=10^(-1:2), gamma=c(.5,1,2)))

print(svm_tunePOT)

svm_model_after_tunePOT <- svm(clPOT ~ ., data=trainset, kernel="radial", type = "nu-regression", cost=10, gamma=1)
summary(svm_model_after_tunePOT)
## CHECK ON TESTSET
predPOT <- predict(svm_model_after_tunePOT,testset)
summary(predPOT)

RMSE_SVMPOT <- rmse(as.numeric(predPOT),checkPOT)
MAE_SVMPOT <- mae(as.numeric(predPOT),checkPOT)


#tuned SVM scores worse then "standard-3" prediction, but close.
```