|  |  |
| --- | --- |
| Presentation date: | 21.06.2024 |
| Topic: | **Machine Learning Algorithms to Predict Employee Turnover** |
| Course: | Machine Learning |
| Lecturer: | Elizabeth Staegemann |
| Data source: | Kaggle  [Employee Turnover (kaggle.com)](https://www.kaggle.com/datasets/davinwijaya/employee-turnover/data) |
| Data type used: | Csv, ipynb, python file |
| Presenters: | Qin, Jinwen  Hrioua, Jad  Serrano Redondo, Antonio  Madjet, Mohamed El-Amine  Mukhitdinova, Lina |

**Table of contents:**

1. Introduction
2. Goal of the project
3. Data preparation
4. Describing the data

**Introduktion:**

According to a study, losing a single employee can cost tens of thousands of dollars, meaning a wave of employee turnover can quickly escalate costs into the hundreds of thousands. Additionally, a newly hired employee may take one to two years to reach the productivity level of an existing employee. Therefore, companies can’t afford such a financial or productivity hit. Companies spend money and time recruiting talented people, and if some of them decide to leave, it represents a loss of investment. Consequently, companies can save money if they can intervene before their employees leave. In fact, measuring employee turnover can be helpful to employers who want to examine reasons for turnover or estimate the cost-to-hire for budgeting purposes.

**Client:** (Auftragsgeber): The HR department wants to know and understand why employees are resigning. They collected data from employees, but now they don’t know what to do with it. They asked us to provide data-driven suggestions based on our understanding of the data.

They have the following question: What’s likely to make the employee leave the company?

The goals in this project are to analyze the data collected by the HR department and to build a model that predicts at 80 % whether or not an employee will leave the company.

The accuracy has been chosen as being a straightforward metric to understand and communicate to stakeholders.

It measures the proportion of correct predictions (both true positives and true negatives) out of all predictions made.

We have a Balanced Dataset and in the case the dataset is balanced (i.e., the number of employees who resign is roughly equal to those who stay), accuracy can be a meaningful measure of performance. In such cases, accuracy will provide a good indication of the model's overall performance.

If we can predict employees likely to quit, it might be possible to identify factors that contribute to their leaving.

**Goal of the project**: By trying different machine learning models, we want to find out which model best predict whether an employee leaves or not and determine the factors which affect his decision making.

**Hypothesis:** *Why are people resigning?*

The most common reasons for resigning from a previous job are:

* Lack of career development/advancement
* Inadequate compensation
* Lack of meaningful work
* Lack of work flexibility
* Employees looking for better opportunities
* Bad management.
* Excessive working hours.
* No promotion
* Not satisfied with the current job
* A negative working environment.

Our objective is to get a score around 80 %, for this we tried to use the following machine learning algorithms starting with supervised ones, as our study deals with classification:

* Bayes (**Lina**)
* Logistic Regression (**Mohamed**)
* Random Forest (**Jad**)
* K-Nearest Neighbors algorithm (KNN) (**Jinwen**)
* Support Vector Machine (SVM) (**Jinwen**)
* Multilayer Perceptron (MLP) (**Jinwen**)
* Decision Tree (**Antonio**)

**Investigation Steps:**

* **Load the corresponding dataset** 
  + Analyze the dataset (looking for outliers, checking missing values, which columns are categorial, and which are numeric, which are feature and which are target).

Since our dataset contains categorical variables, we need to convert them into numerical representations that can be understood by the model.

-Which columns are not relevant to the current study or can’t provide useful information.

-We checked also the dependence between different features.

-Check which categories can be combined (see mapping.xls)

-We considered all the columns, hoping that the algorithms can find useful features.

**Data preparation**

1. First, pandas is used to load the ‘turnover.csv’ dataset: df = d.read\_csv('turnover.csv', encoding = 'ISO-8859-1');
2. The dataset has 1129 rows (employees) and 16 columns, no missing values was detected; The dataset looks clean and there are no missing values;
3. df.describe() provides also mean values and for ‘event’ one gets 50.6 %.

Balanced dataset: 50.6 % of the employees left the organization and 49.4 % did not leave the organization making our dataset to be considered as balanced.

1. Categories combining and Encoding

For the columns industry, profession, coach and traffic we combined categories.

We also used two encoders:

One Hot encoder: industry, profession, traffic, coach

Label encoder: way

The rest of the features were already encoded.

The new dataset is saved in a new file: 'df2\_encoded.csv', which is used for different machine learning algorithms.

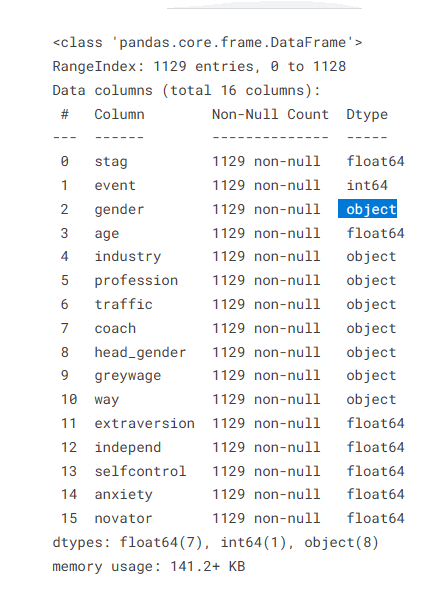
From this step, each of us use this dataset and test the corresponding machine learning model.

1. Scaling We used two approaches: first without scaling the data and the second approach is scaling the data with both StandardScaler and MinMaxScaler.

**Describing the data**

**Columns Attributes**

We have the following attributes column-wise:



* stag - Experience (time) in months
* event - Employee turnover, this hast two possible values: 1 for resigning and 0 for staying
* gender - Employee's gender, female(f), or male(m)
* age - Employee's age (year)
* industry - Employee's Industry
* profession - Employee's profession
* traffic: From what pipeline employee came to the company:

1. KA: The recruiting agency brought you to the employer.
2. Advert: contacted the company directly, after learning from advertising.
3. Empjs: The employer contacted the candidate after seeing his resume on the job site.
4. Friends: The employer contacted you on the recommendation of a person who knows you
5. rabrecNErab: the employer contacted a candidate on the recommendation of a person who knows him.
6. recNErab: contacted the company directly on the recommendation of your friend - NOT an employee of this company.
7. Referal: contacted the company directly on the recommendation of your friend - an employee of this company
8. youjs: You have applied for a vacancy on the job site

* **Coach**: Presence of a coach (training) during probation. This categorical attribute has the following values: (Yes/ No/ my head)
* head\_gender: head (supervisor) gender: male or female.
* greywage: The salary does not seem to the tax authorities

white: the employee becomes only salary,

grey: the employee gets salary and other benefits (car, phone..)

* way: Employee's way of transportation.

1. Bus: takes the bus to go to work.
2. Car: the employee drives to work with his car.
3. Foot: the employee goes by foot to work.

* **Extraversion**: personality trait characterized by excitability, sociability, talkativeness, assertiveness, and high amounts of emotional expressiveness.
* **Independent**: the ability to make and carry out important decisions, without outside coaching.
* **Selfcontrol**: The ability to control behaviors to avoid temptations and achieve goals.
* **Anxiety**: People who tend to experience apprehension, tension, or uneasiness that stems from the anticipation of danger, which may be internal or external.
* **Novator**: An innovator.

We differentiate between the following attributes of the dataset and their type

**Categorial**: gender, industry, profession, traffic, coach, head, gender, greywage, way

**numerical**: stag, event, age, extraversion, independent, selfcontrol, anxiety, novator

**Observations/ hypotheses about the dataset:**

**Age**

**Category age,** taking a subset of the age range:

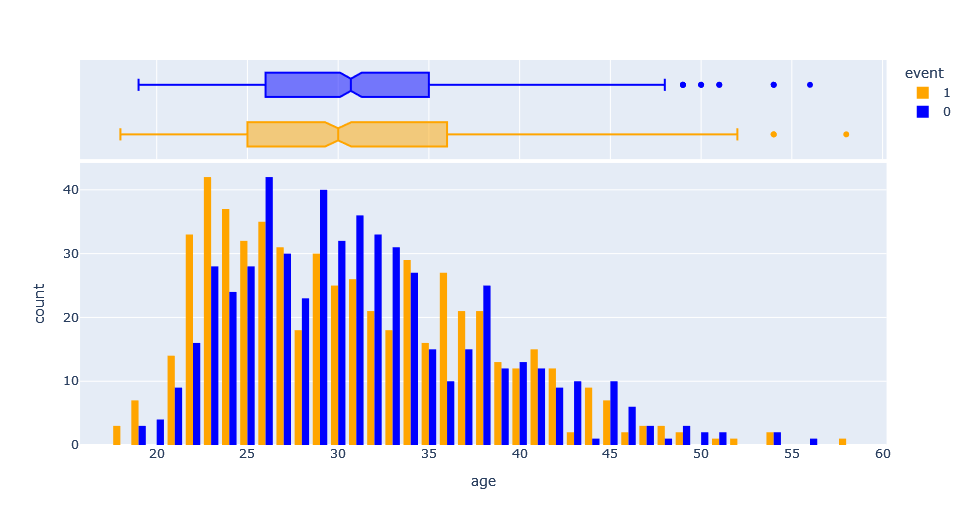
* **first category**: from 18 to 25 years.

the reason why employees from this category are resigning is due to unsatisfaction with the Chef/Coach or the employees are not happy with the choice they made for the training/Ausbildung.

* **the second one** is from 26 to 36 years:

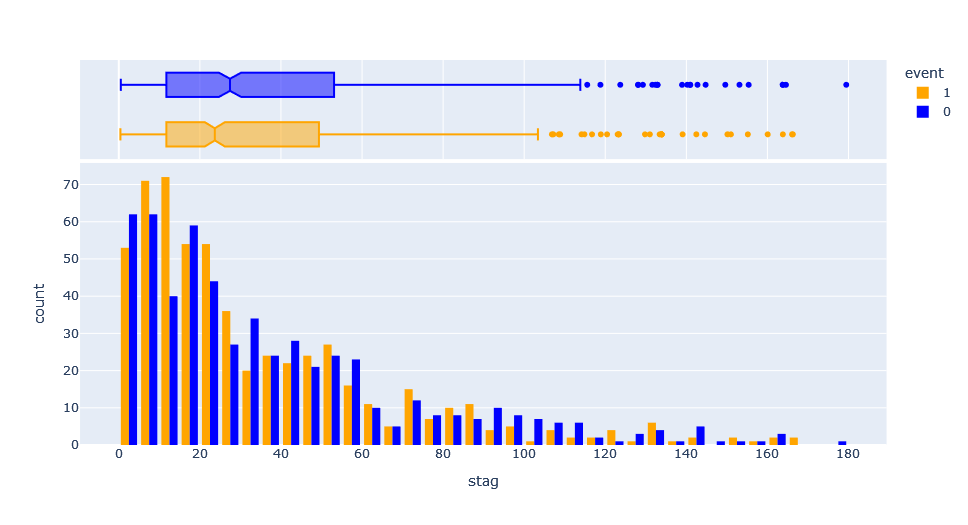
most of the employees are staying as they want probably to gain more expertise and hope for promotion.

* **the third range** is from 37 to and 45 years: half decided to stay, and half left the company. Maybe the employees who left are looking for more perspectives and more challenges.
* **the fourth one** is for those older than 45 years, at this age, the employees are looking for stability and they prefer to stay rather than leaving the company as the chances of getting a new job is very limited.



**Stag –** describes the time spent in the company and it is given in months

* **From 0 to 5 moth**: probation period.
* **From 6 to 30 months**: fixed term contract, not happy with the new job.
* **From 31 to 59 months:** after gaining expertise, employees are leaving after looking for a better paid job and with more challenges.
* **From 60 to 90 months**: after 5 years of work in a company, some employees who are not getting promoted tend to resign.
* **Between 90 and 120 months:** after more than 7 years of employment, most employees prefer to stay; seeking stability.



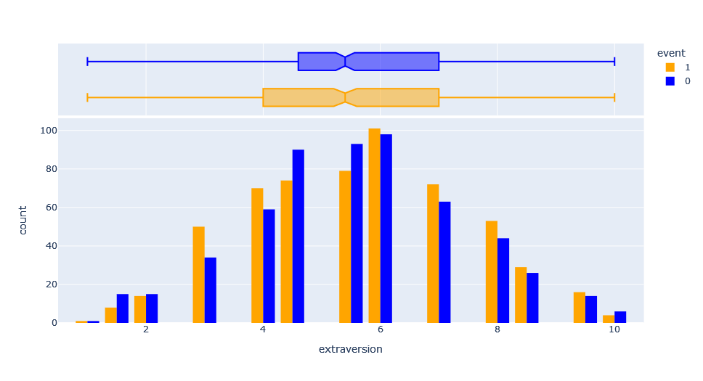
**In our opinion: What is missing from the dataset to identify the main reason why the employees are leaving their job:**

* Salary amount & Working number of hours per week -> could contribute to burnout;
* Position: senior, junior, manager or trainee;
* Business trip/ hybrid or Working from home possibilities;
* Possibility for promotion or whether the employee was promoted in the past;

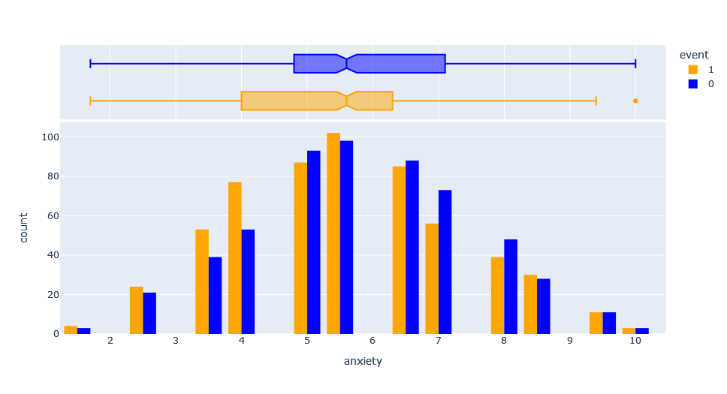
**Selfcontrol, Novator and Independent features seem to mimic the Normal distribution:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
| A graph with different colored bars  Description automatically generated | |  | |

**Extraversion** – for extraversion value less than 4.6, more employees left compared to those who decided to stay.

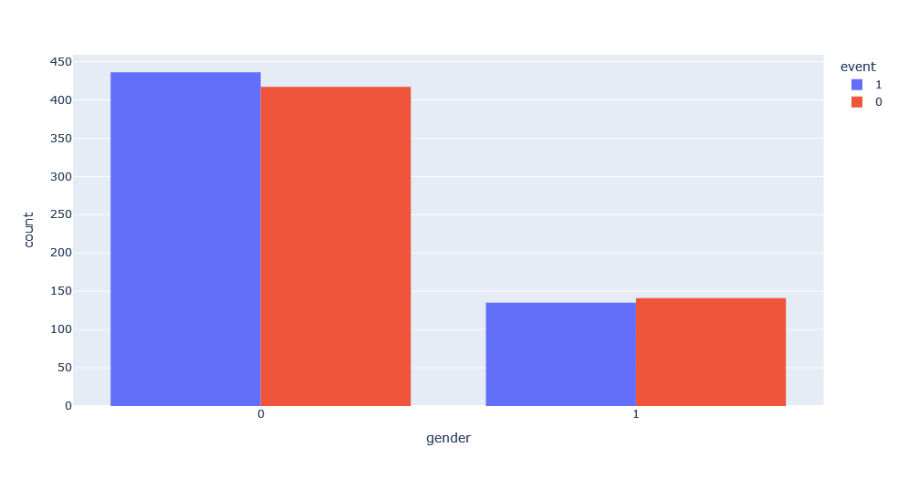


**Anxiety** - for an anxiety value between 0 and 4.8, more employees decided to leave, taking the risk to look for a new job. Then, for a value ranging from 4.8 to 6.3, half of the employees resigned. For a value greater than 6.3, employees prefer stability as they are scared of not finding a new job if they decide to leave the company, feeling they are in a comfort zone. This shows a direct correlation between leaving/staying and anxiety, and therefore, this column is important to consider in our study.



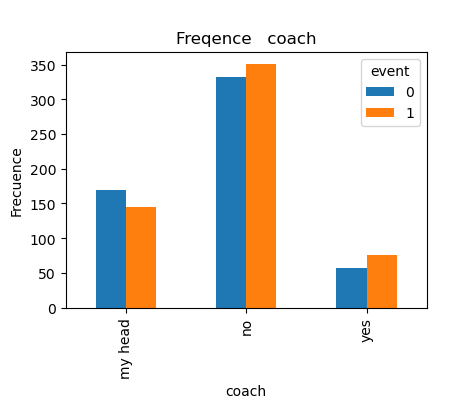
Now, let’s extend our investigation to other columns:

**Gende**r: As we can see on the following figure, that almost 50% of man and woman resigned and the same percent stayed.

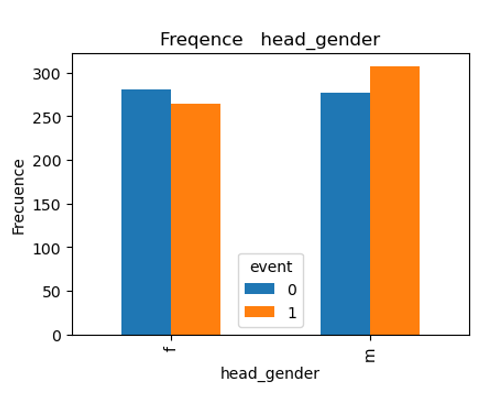


By looking at this graph we don’t get additional information. However, we should consider it in our analysis as the other features could be linked to this one.

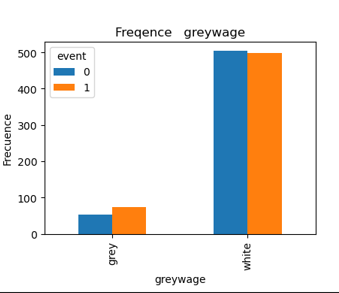
**Coach:** We can reduce these one to two possibilities, yes (by combining my head and yes) and no.



**Head gender:**



**Grey wage:**



**Intensive Testing every Model**

**For all the models the data was divided into Training and Testing sets:**

It is crucial to evaluate our models’s performance on unseen data to assess their ability to generalize. We use the train\_test\_split() function to split the data into training and testing sets. The X\_train, X\_test, y\_train, and y\_test variables store the resulting split, with 80% of the data used for training and 20% for testing.

Tr=Training, te=testing

show

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | BGs | DT | RF | LR | KNN | SVC | MLP |
| trainingsscore | 0.58(tr) | 1 (tr) | 0.65 (te)  1 (tr) | 0.59(tr) | 0.74(tr) | 0.58 | 0.76 |
| crossvalscore | 0.53(tr) | 0.61 (tr) | 0.63 (tr) | 0.56(tr) | 0.58(tr) | 0.55 | 0.58 |
| Grid-SearchCV | 0.58(tr) |  | 0.64 | 0.59(tr) | 0.64(tr) | 0.55 | 0.59 |
| Score auf Testmenge | 0.53 |  | 0.588t(te) | 0.56(te) | 0.66(te) | 0.54 | 0.55 |

**Fine Tuning Model**

|  |  |
| --- | --- |
| Decision Tree (Antonio) | * Fit with default: no Scaling is best * Cross\_Val\_Score :8 folder is best * GridSearchCV: default values are the best * On Test Menge: 0.55 |
| GaussianNB, BernoulliNB (Lina) | * Fit with default: MinMax , Standard or no scaler does not matter for Bayes * Tried PCA – did not improved the score; * Tried 3 types of Bayes: Gauss, Bernoulli, * Cross\_Val\_Score :see below * GridSearchCV: variance\_smoothing * On Test Menge: 0.53 |
| RandomForest(Jad) | * Fit with default: **MinMax Scaling is best** * Cross\_Val\_Score :9 folder is best * GridSearchCV: {'bootstrap': False, 'criterion': 'entropy', 'max\_depth': None, 'max\_features': 6, 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 30   }   * On Test Menge: 0.588 |
| LogReg(Mohamed)  + Interaction terms | * Fit with default: MinMax **Scaling is best** * Cross\_Val\_Score :3 folder is best * GridSearchCV: {'C': 1, 'penalty': 'l2', 'solver': 'saga'} * On Test Menge: 0.56 * On Test Menge: 0.62 |
| KNN(Jinwen) | * Fit with default: **no Scaling is best** * Cross\_Val\_Score :8 folder is best * GridSearchCV: {'algorithm': 'auto', 'leaf\_size': 10, 'metric': 'manhattan', 'n\_neighbors': 9, 'weights': 'distance'} * On Test Menge: 0.66 |
| SVC(Jinwen) | * Fit with default: **no Scaling is best** * Cross\_Val\_Score :7 folder is best * GridSearchCV: Best parameters: {'decision\_function\_shape': 'ovo', 'kernel': 'linear'} * On Test Menge: 0.54 |
| MLP(Jinwen) | * Fit with default: **Standard Scaling is best** * Cross\_Val\_Score :7 folder is best * GridSearchCV: Best parameters: {'activation': 'tanh', 'alpha': 0.0001, 'hidden\_layer\_sizes': (30,), 'learning\_rate': 'constant', 'solver': 'adam'} * On Test Menge: 0.55 |

# **Logistic Regression model (Mohamed)**

### **Introduction**:

Classification is an area of supervised machine learning that tries to predict which class or category some entity belongs to, based on its features. Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. **This ML algorithm is a technique that models a categorical dependent variable (Y) based on one or more independent variables(X).** Logistic regression uses the concept of predictive modeling as regression; therefore, it is called logistic regression, but is used to classify samples; Therefore, it falls under the classification algorithm.Logistic regression predicts the output of a categorical dependent variable. Consequently, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

### **Logistic Function (Sigmoid Function)**

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the **Sigmoid function** or the logistic function.

### **Assumptions for Logistic Regression**

1. The dependent variable must be categorical and binary.
2. There should be little or no multicollinearity between the independent variables.
3. There is linear relationship of variables to log odds.
4. Requires sufficiently large sample size.
5. There are no extreme outliers.
6. Should have independent observations.

### **Steps in our Logistic Regression model**:

1. Import packages, functions, and classes
2. Used non scaled data to compute the scores on training and test data, we got respectively, 0.58 and 0.56.
3. Used training Data with Standard scaling and 0.58 and 0.55 for training and testing data, respectively.
4. Used MinMax scaling and got 0.59 and 0.56, respectively for training and testing scores.
5. MixMax scaling seems to pride slightly better scores, therefore this scaling was used
6. For Cross Value score, we found 3 folds to lead to the best score
7. Using gridSearch we identified the best parameters for the model as: Hyperparameters: {'C': 1, 'penalty': 'l2', 'solver': 'saga'}, we obtained the following training and test score: 0.59 and 0.56, respectively. In general, using different options for the model had almost no pronounced effect on improving the scores. Maybe this has to do with the nature of our dataset.
8. We calculated the coefficients sorted by the magnitude of each feature's impact on employee turnover. In Absolut value, age, anxiety and stag have the largest coefficients.
9. Incorporated the interaction terms into the logistic regression model to assess whether this enhancement leads to improved model performance. We introduced all possible interactions, this extensive set of interactions could lead to overfitting, making the model too specific to the training data and potentially hindering its generalization to new, unseen data. Maybe by strategically selecting a subset of interactions, we can maintain model interpretability and efficiency while capturing the essential nuances in the relationship between different predictors and event (employee resigning). If we had more time we could investigate more this by selecting the interactions terms.

**Lessons Learned (Mohamed)**

* The teamwork was a great experience, we had interesting discussions during the last three days, exchanged ideas and learned from each other.
* I gained experience from this project, and learned how to prepare the data in a format that is more appropriate for machine learning algorithms. I learned how to use scaled and non-scaled and scaled data and compute score on test and train data, how to improve the parameters of the ML model.
* The ‘event’ in the dataset, has a weak correlation with other features which makes probably hard to get high scores. In fact, the dataset, we selected seems was not the best one, as it is missing many important features which could directly impact the employee resignation, like salary, working hours, satisfactions, promotion.
* In my opinion, we need more time to conduct a project on Machine Learning and get a deep understanding, and to tune correctly the different parameters of the model. For example, for the logistic regression model, it seems including the interaction terms lead to better score, but I did not get enough time to incorporate selective interaction terms.

1. **Random Forest Classification Modell (Jad)**

Random Forest ClassificationModell

für Turnover Dataset

**Einführung**

Random Forest ist ein Ensemble-Lernalgorithmus, der mehrere Entscheidungsbäume trainiert und deren Vorhersagen kombiniert

A diagram of a diagram

Description automatically generated

Bei Klassifikation wird der Wert am **häufigsten** die von trees ausgegeben ausgewählt.

A diagram of a tree

Description automatically generated

**Erledigte Arbeit**

Nach der Bereinigung der Daten und Encoding der Kategoriale Features

werden die Daten in einer csv Datei gespeichert.

Schritte um das beste Modell zu finden.

* Der Model wird **ohne Skalierung** trainiert und die Scores für Trainings- und Testdaten ermittelt
* Der Model wird mit **Standard Scaling** trainiert und die Scores für Trainings- und Testdaten ermittelt
* Der Model wird **mit MinMax Scaling** trainiert und die Scores für Trainings- und Testdaten ermittelt
* **Fit with default: MinMax Scaling is best**
* **Cross-Validierung** wird mit verschiedenen Folds zwischen 2 und 10 durchgeführt und Scores verglichen
* **Cross\_Val\_Score :9 folds is best**
* Hyperparameter werden mit **GridSearchCV** optimiert für den besten Estimator mit dem besten score
* **GridSearchCV: {'bootstrap': False, 'criterion': 'entropy', 'max\_depth': None, 'max\_features': 6, 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 30 }**
* Überprüfung der Score mit dem best\_Estimator
* **On Test Menge: 0.588**

**Ergebisse**

|  |  |
| --- | --- |
|  | Random Forest |
| trainingsscore | 0.65 (Testmenge)  1 (Trmenge) |
| crossvalscore | 0.63 (Trmenge) |
| Grid-SearchCV | 0.64 |
| Score auf Testmenge | 0.588(Testmenge) |

* Fit with default: MinMax Scaling is best
* Cross\_Val\_Score :9 folds is best
* GridSearchCV: {'bootstrap': False, 'criterion': 'entropy', 'max\_depth': None, 'max\_features': 6, 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 30 }
* On Test Menge: 0.588

**Erkenntniss:**

**"event" im Datensatz hat eine schwache Korrelation mit anderen Merkmalen, was es wahrscheinlich schwierig macht, hohe score zu erhalten. Tatsächlich scheint der von uns ausgewählte Datensatz nicht der beste zu sein, da viele wichtige Merkmale fehlen, die sich direkt auf die Kündigung eines Mitarbeiters auswirken könnten, wie z. B. Gehalt, Arbeitszeiten, Zufriedenheit und Beförderung.**

**Schlussvolegeung**

Random Forest ist eine Technik, die Ensemble-Lernen verwendet hat aber Vorteile und Nachteile

* **Vorteile**
* Es ist vergleichsweise langsamer.
* RandomForest bietet eine eingebaute Methode zur Bewertung der **Importance** von Features, was bei der Merkmalsauswahl und der Interpretation des Modells hilfreich ist
* **Parallelisierbarkeit** Die Berechnung der einzelnen Entscheidungsbäume ist voneinander unabhängig, was eine parallele Verarbeitung und damit eine schnellere Modellierung ermöglicht.
* Robust gegen Rauschen
* **Nachteile**
* Rechenaufwand und Speichernutzung: viele Trees gleichzeitig trainiert und gespeichert werden müssen
* Overfitting entgegenwirkt
* Langsame Vorhersagegeschwindigkeit

**Lessons Learned**

* Schwierigkeiten alle Parameter zu verstehen. Das könnte helfen bei der Optimierung von den Hyperparametrer.
* Um Daten optimal vorzubereiten braucht man viel Zeit für die Analyse der Daten
* Es wäre sinnvoll bei nächsten Projekten regelmäßig zu treffen um über Fortschritte und Probleme auszutauschen
* GridSearch hat bei mir lange gedauert wegen der großen Anzahl der Parameter und wie den Algorithmus arbeitet
* wenn man mehr Zeit hätte könnte man die Kategorien anders zusammenfassen

1. **Naïve Bayes Classifier** (Lina Mukhitdinova)
2. Differences of 3 Bayes classifiers:\*
3. **Bernoulli NB**: If we consider a dataset is\_extraverted, is\_inselfcontrol, is\_novator – you classify the person as resigning or not (in the context of the dataset at hand)
4. **GaussianNB**: if we consider a dataset which is normally distributed.

* **Best score of 0.58**

1. **ComplementNB**: is particularly suited for imbalanced datasets

* **Poor score of 0.55**

I investigated the above three cases but there are more.

1. **Additional data cleaning**: if we consider the correlation matrix below we can see the following personal traits are negatively correlated:

* Corr(Selfcontrol , novator) = -0.6
* Corr(Selfcontrol , extraversion) =-0.5
* Corr(independent, anxiety) = -0.4

A colorful squares with black text

Description automatically generated

Dropped features which correlate:

features1 = df.drop(['event','extraversion','independ','novator' ,axis=1)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features1, target, test\_size=0.2, random\_state=42)

With this parameters dropped:

Training Score Gauss: 0.5858250276854928,

Crossval\_score Gauss:0.5359931170108162



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \*dropped parameters | df.drop(['event','extraversion','independ','novator','gender' ],axis=1) | | |  |  |
| \*\* default data | drop(['event'],axis=1) |  |  |  |  |
|  |  |  |  |  |  |

1. **Specifically for NBBernoulli** – tired to transform continuous variables “continuous” parameters extraversion, independ, selfcontror, anxiety and novator (all within the range from 1 to 10 in the original dataset) to classify whether, i.e.:

if\_independent =Yes/No (Yes – if rated above 5, No -if a person rated him-/herself below 5)

|  |  |
| --- | --- |
| * Poor results did not improve the score |  |

1. **Additional observations/remarks on the data set:**
2. In the original dataset ca. 75% of respondents are females and most of the respondents work at HR:

* Typically, HR roles are female dominant – would ask the client what is the goal of this prediction & whether it is directed only to people who work in an HR department.

A graph with blue and orange bars

Description automatically generatedA graph with numbers and a bar chart

Description automatically generated

* Would ask for additional data – hours per week, or number of projects assigned, if promoted in the last 3 years, etc.

Lessons Learned (Lina)

* Schwierigkeiten alle Parametern zu verstehen und was diese mit dem Model machen (mathematisch gesehen und auch Auswirkungen auf das Endergebnis)
* Projektarbeit: würde das Berichtschreiben zuerst einer Person überlassen und dann sobald fertig ist, Details als Gruppe korrigieren.
* Generell: man braucht mehr Zeit um sich zum ersten mal mit Machine Learning anzufreunden - brauche meine Zeit um den Stoff zu verstehen und verdauen – statt gleich dieses neue Thema mit der Gruppe zu diskutieren;
* Was ich zusätzlich machen würde: würde den code als Klassen und auch funktionen hinzufügen schreiben, zusätzliche Methoden aussuchen

1. **Churn Projekt: CODE ERKLÄRUNG: DECISSION TREE (Antonio)**

Das ziel dieses Dokuments ist eine Zusammenfassung des Codes mitteilein. Themen, die wichtig über dem Code sind, auch in “Lessons learns“ besprochen werden.

Der Modell ist ein Decission Tree Classifier, (wir benutzen nicht Regression weil unsere Ziel Spalte ist eine Kategorie -Ja, Nein-).

Der Wähl liegt in die Möglichkeit, dass Decission Tree hast um mit Kategorien und Werte zu arbeiten.

Algorithmus Vorgehen:

1. Data Vorbereitung (allgemein für die Gruppe, steht in Gruppe Bericht)
2. Split in Training Data und Test Data. Hier wir nehmen 80% Training Data und 20% Test Data.

Y Spalte : Event (Mitarbeiter kündigt)

X Spalte : Feature Spalten

1. Wir trainieren die Daten und machen wir ein Prognose, um zu sehen wie gut der Algorithmus ist. Wir bekommen 1 in Trainings Data. Das bedeutet, dass unsere Modell sehr gut mit dem Trainierte Data rechnen kann, aber das bedeutet nicht, dass mit neuen Data sehr gut rechnen auch kann. Wir überprüfen es mit dem Test Data, und hier haben wir ein score 0.64.

Wir können das bei Decission Tree erwarten, ein Overfitting bei Training Data zu bekommen. Deswegen, um zu eigentlich prüfen, dass der Modell mit “hidden data“ gut rechnen kann, versuchen wir Cross Validierung Score in die Training Data.

Wir rechnen Cross Val. Score in Training Data und wir merken, dass die beste Folds Nummer ist 8, weil die Beste score hast mit sehr wenig Standard Deviation. Deswegen benutzen ab diesem Moment immer 8 Folds.

4. Optimierung.

1. **Features Wichtigkei**t: Wir nehmen die Spalte die unwichtig sind raus. Der Modell rechnet sie Wichtigkeit jeder Spalte (summe Spalte ist 1). Damit können wir wissen, welche Spalten raus dürfen, ohne wichtige Information zu verlieren. Wir haben gesetzt wie Grenze 2%. Das bedeutet, die Spalte deren Wichtigkeit weniger als 2% ist, werden herausgefiltert. Mit diesem Kriterium, 5 Spalten sind Weg.  
   ['coach', 'greywage', 'way', 'industry', 'gender', 'profession']

Wir rechnen wieder der Algoritmus mit die Spaltenraus, aber der Score hat sich nicht verbessert. Diesem Schritt der Optimierung war nicht hilfreich.

B) **Skalierung**: Hier versuchen wir die Daten ähnlicher machen, um auch die Modell vereinfachen. Wir versuchen mit wei Optionen, Standard Scale und MinMax Scale.

Wir führen wieder die Cross Val Score auf Training Data us. Leider ist die Skalierung auch nicht hilfreich.

C) **Grid Search**: Wir versuchen mit dem GridSearch das Modell zu verbessern durch eine Parameter Optimierung. Das Default Decission Tree Classifier hat alle Paramenter mit ein Default Option, die verändert werden können.

Das Ziel ist, die beste Parameter zu finden, die einen besseren Score rechnen. Mein Ansatz war mit dem Parameter mitspielen, leider wenn ich versuche verschiedene Parameter Werte zu bieten, crash den Laptop. Deswegen musste ich nur eigene Parameter mit nicht zu viele Werte setzen.  
  
Zum Beispiel bei Criterion, Falls ich die drei Optionen setzen (Gini, Entropy, Log\_loss) sodass der Algorithmus alle Kombinationen versuchen kann, bekomme ich keine Score nach 60mn.  
  
Deswegen mit weniger Parameter Optionen und sehr leicht, wird der Score nur 0.55 werden. Das bedeutet, dass die Default Parameters sind besser als die Grid Search, leider haben wir nicht gefunden, welche wären am besten.

Wir rechnen die “Best parameters” (die wir gerade wissen die nicht die beste sind) auf die Test Data und bekomment wir auch 0.55.

Der Ansatz die Parameter Optimierung war ein kleines Baum zu erstellen, ohne zu viele Nodes und Leafs. Auch die min samples möchten wir, dass nicht zu viele sind.

**LESSONS LEARNT (Antonio)**

**Modell Inhalt**

Was ich am besten gelernt in diesem Projekt habe sind die verschiedene Schritte zu nehmen um ein Final Score zu haben. Der GridSearch war mir nicht ganz klar. Ich dachte es war ein Art von Column Importance aber habe ich jetzt viel besser verstanden, dass die letzte Schritt für einen optimierten Score ist.

Bei Modell, habe ich auch gemerkt, dass Columns Importance kann eigentlich zweideutig sein.

Zum Beispiel hatte ich vor der Code genau schreiben ein bisschen mit der Data mitgespielt. Ich hatte bei verschiede Spalte gefiltered und die mean von Y Spalte kalkuliert. Leider hatte ich keine große Unterschied zwischen die Mean() von Event = 1 und Event = 0. Allerdings bei Profession Commercial gefiltered, hatten in Event 1 die mean() = 7 und bei Event = 0 mean() = 5. Das fand ich eine gute Anmerkung. Ich kann mir vorstellen, dass die Leute die in Commercial arbeiten , haben Sales Ziel jeder Monat und das Gehalt hängt davon ab, deswegen die Leute die mehr Anxiety haben, können damit nicht umgehen und am Ende kündigen.

Deswegen dachte ich, dass es für den Baum Erstellung Profession wichtig sein könnte, allerdings war die Spalte Profession rausgenommen. Das war nur ein kleines Beispiel aber sicher dass es viel mehr gibt, die ich nicht gesehen habe, die raus von Modell wären. Daher habe ich gelernt, dass Decission Tree besser mit spezifische Information arbeiten und nicht mit zu viele Spalten, weil der Baum zu groß wird. Ich frage mich, ob es gibt die Gelegenheit die Importance Column nutzen anders: Nicht um Columns rausfiltern sonst für Spalte-n wie Wurzeln benutzen, um der Baum große zu reduzieren.

Ich habe auch gelernt, wie wichtig ist eine gute Kategorie Verteilung erstellen in der Daten Vorbereitung. Mann muss nicht nur die Strings in Zahlen umsetzen. Die Zählen Gruppierung sollten eine Verbindung mit der Realität haben. Ein Beispiel in unsere Gruppe war die Gruppierung von Spalte “Way”. Wir haben By Foot, Car, Bus. Die Gruppe hat es interpretiert wie der Distanz von der Arbeit und dachten, dass eine angemessene Gruppierung wäre By Foot = near, Bus = middle, Car = far. Ich dachte dass es besser wäre, nur 2 Gruppe machen: By Foot = Without vehicle und Bus - Car = With vehicle (Meine Grund war, dass vielleicht die Jungen unabhängig von der Distanz den Bus einfach nehmen, weil sie kein Geld haben, um ein Auto zu kaufen). Beide Interpretationen der Daten sind gut aber jeder Option hat ein Einfluss in der Finale Baum. Der Lesson ist: Die Interpretation der Features in der Realität, dass wir analysieren möchten, hat eine sehr große Einfluss in der Ergebnisse. Deswegen lohnt es sich, genug Zeit nehmen zu überlegen in der Daten Vorbereitung.

Was ich leicht fand ist ein Score zu bekommen. Es ist relativ einfach mit dem Library einen Score zu rechnen. Allerdings ist es sehr schwierig, der Score zu optimieren. Ich habe auch gemerkt, dass es sehr hilfreich wäre, besser die Parameter Inhalt zu lernen. Daher hätte man mehr Gelegenheiten, es zu optimieren.

Es war auch sehr schwierig der Projekt ausführen in 3 Tage, vielleicht nicht genug Zeit dafür. Insbesondere für Leute ohne Erfahrung in Machine Learning.

1. **KNN, SVC (Jinwin)**

KNN

KNN, which stands for K-Nearest Neighbors, is a simple yet powerful algorithm used in machine learning for classification and regression. Here's a concise introduction:

Ein Bild, das Kreis, Diagramm enthält.

Automatisch generierte Beschreibung

1. \*\*Concept\*\*: KNN is a non-parametric method that relies on the idea that similar data points are likely to have similar labels. It classifies new data points based on the majority class of its 'k' nearest neighbors in the feature space.

2. \*\*How it Works\*\*:

- \*\*Distance Measurement\*\*: The algorithm calculates the distance between the new data point and all the points in the training set. Commonly used distance metrics include Euclidean, Manhattan, and Hamming distance.

- \*\*Choosing 'k'\*\*: The value of 'k' determines the number of nearest neighbors to consider. It's a hyperparameter that can be tuned.

- \*\*Majority Voting\*\*: For classification, the algorithm assigns the class that is most common among the 'k' nearest neighbors. For regression, it might take the average of the values of the 'k' nearest neighbors.

3. \*\*Advantages\*\*:

- Easy to understand and implement.

- Does not require training time as it uses the entire dataset for classification.

- Can be effective with a good choice of 'k' and distance metric.

4. \*\*Disadvantages\*\*:

- Computationally intensive as it requires calculating the distance to all points in the dataset.

- Sensitive to irrelevant features and the scale of features.

- Requires careful selection of 'k' and distance metric.

5. \*\*Applications\*\*: KNN is used in a variety of applications, including recommendation systems, image recognition, and anomaly detection.

6. \*\*Implementation\*\*: It can be easily implemented in many programming languages and machine learning libraries, such as Python's scikit-learn library.

KNN is a versatile algorithm that, despite its simplicity, can be very effective for certain types of data and problems.

A blue arrows pointing to different directions

Description automatically generated

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

SVC

Support Vector Machine (SVM) is a powerful and versatile supervised machine learning algorithm used for both classification and regression tasks. Here's a brief introduction:

Ein Bild, das Reihe, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

1. \*\*Concept\*\*: SVM is based on the idea of finding the optimal hyperplane that best separates data into different classes. The goal is to maximize the margin between the closest points of the classes, which are known as support vectors.

2. \*\*How it Works\*\*:

- \*\*Hyperplane\*\*: In a two-dimensional space, a hyperplane is a line that separates two classes. In higher dimensions, it's a hyperplane.

- \*\*Margin\*\*: The gap between the hyperplane and the nearest data points from each class is the margin. SVM aims to maximize this margin.

- \*\*Support Vectors\*\*: The data points that lie closest to the hyperplane and influence its position and orientation are called support vectors.

3. \*\*Kernel Trick\*\*: SVM can handle non-linearly separable data by using kernel functions, which allow the algorithm to operate in a higher-dimensional space without explicitly computing the coordinates in that space. Common kernels include linear, polynomial, radial basis function (RBF), and sigmoid.

4. \*\*Soft Margin\*\*: To handle noisy data, SVM introduces a soft margin that allows some misclassifications in exchange for better generalization on unseen data.

5. \*\*Advantages\*\*:

- Effective in high-dimensional spaces.

- Performs well with a clear margin of separation.

- Versatile due to the kernel trick.

6. \*\*Disadvantages\*\*:

- Can be memory-intensive with large datasets.

- Choice of kernel and its parameters can significantly affect performance.

- May not perform well if the data is not linearly separable without careful parameter tuning.

7. \*\*Applications\*\*: SVM is widely used in various applications, including image classification, bioinformatics, text categorization, and handwriting recognition.

8. \*\*Implementation\*\*: SVM can be implemented using various libraries and frameworks, such as scikit-learn in Python, which provides a straightforward way to apply SVM to different datasets.

In summary, SVM is a robust algorithm that can handle both linear and non-linear data separation by maximizing the margin between classes, making it a popular choice for a wide range of machine learning tasks.

A blue arrows pointing to a black background

Description automatically generated

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

MLP

The Multi-Layer Perceptron (MLP) Classifier is a type of artificial neural network that is widely used for classification tasks. Here's a brief overview:

Ein Bild, das Kreis, Diagramm, Screenshot enthält.

Automatisch generierte Beschreibung

1. \*\*What is it?\*\*: An MLP is a feedforward artificial neural network that consists of multiple layers of neurons, including an input layer, one or more hidden layers, and an output layer.

2. \*\*How it works\*\*:

- \*\*Input Layer\*\*: Receives the feature vectors of the input data.

- \*\*Hidden Layers\*\*: These layers perform computations using weights and biases, and introduce non-linearity through activation functions.

- \*\*Output Layer\*\*: The final layer, which uses an activation function appropriate for the task (e.g., softmax for multi-class classification).

3. \*\*Activation Functions\*\*: Commonly used functions include ReLU for hidden layers to introduce non-linearity and softmax for the output layer to normalize output values into probabilities.

4. \*\*Training Process\*\*:

- The network is trained using labeled data by adjusting the weights and biases through backpropagation and an optimization algorithm like gradient descent.

5. \*\*Advantages\*\*:

- Capable of learning complex patterns and relationships in data.

- Flexible architecture allows for the addition of more layers and neurons to increase model complexity.

6. \*\*Disadvantages\*\*:

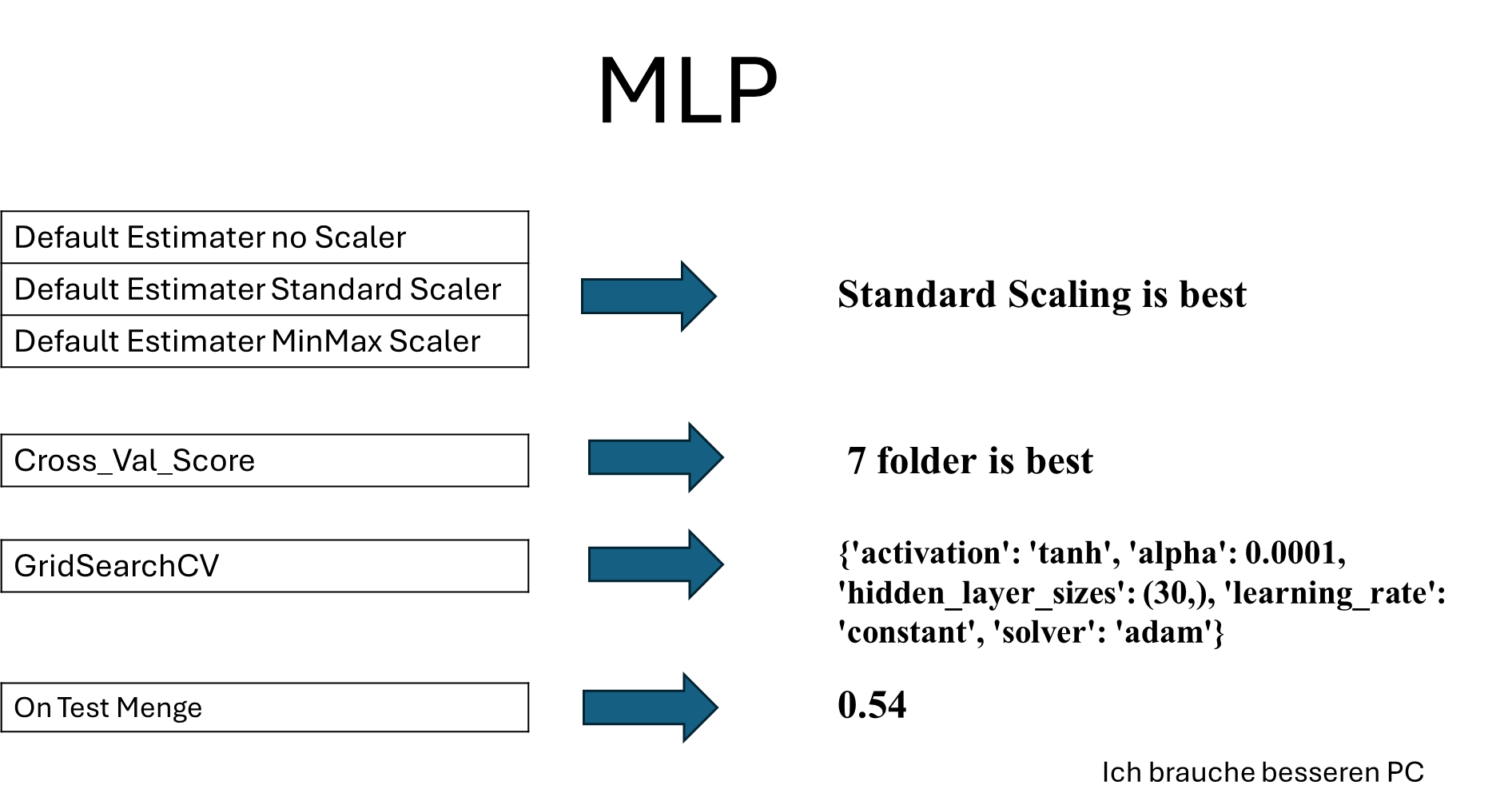
- Prone to overfitting if not regularized properly.

- Requires careful hyperparameter tuning, including the number of layers and neurons.

7. \*\*Applications\*\*: MLPs are used in various applications, including image recognition, speech recognition, and natural language processing.

8. \*\*Implementation\*\*: MLP Classifiers can be easily implemented using machine learning libraries such as TensorFlow, Keras, or PyTorch.

In summary, the MLP Classifier is a powerful tool for classification tasks, capable of learning from data and making predictions based on complex patterns.



Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

All Score

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Leasons Learned (Jenwin)

was gut gemacht wird:

\*Teamzusammenhalt

\*Jedes Teammitglied war für seine Arbeit verantwortlich

\*Der Tutor hat auf Fragen geantwortet, wenn wir unsicher waren.

\*Wir konnten Erfahrungen für das erste Projekt mit maschinellem Lernen sammeln

was nicht gut gemacht wird:

- Ich verstehe die Parameter der einzelnen Modelle nicht genau. Ich kann die Feinabstimmung des Modells nicht sehr gut vornehmen.

- Es ist schwierig anzufangen, wenn alle zum ersten Mal ein Projekt zum maschinellen Lernen durchführt.

Was würde ich tun, wenn ich mehr Zeit und einen guten PC hätte?

\*PCA-Ergebnisse ausprobieren und umsetzen

\*mehr über die Parameter jedes Modells lernen. auf GridseachCV mehr ausprobieren.

\*Mehr Kategorie Zusammenfassung ausprobieren