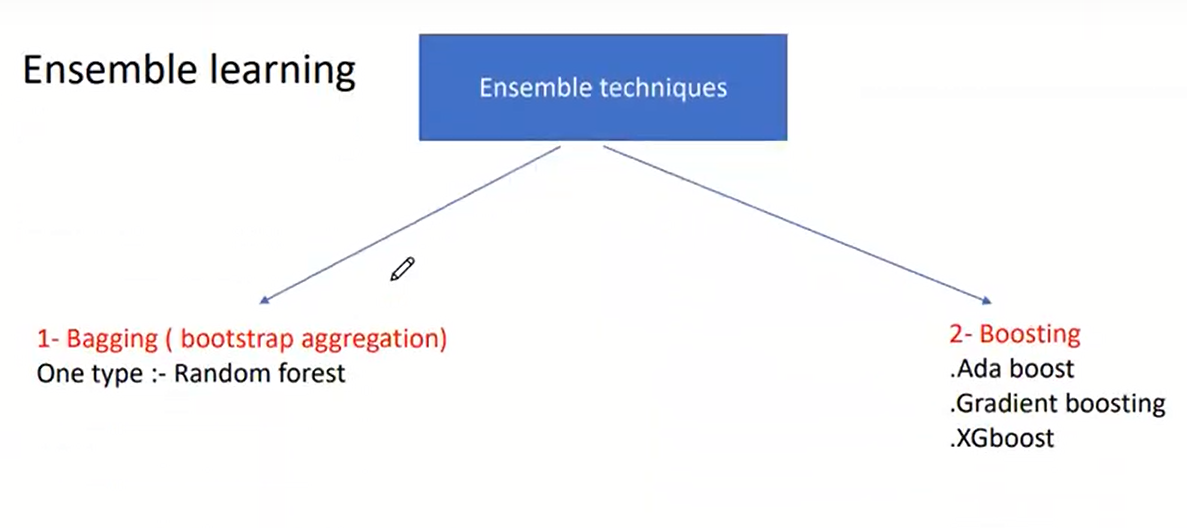
Random Forest ClassificationModell

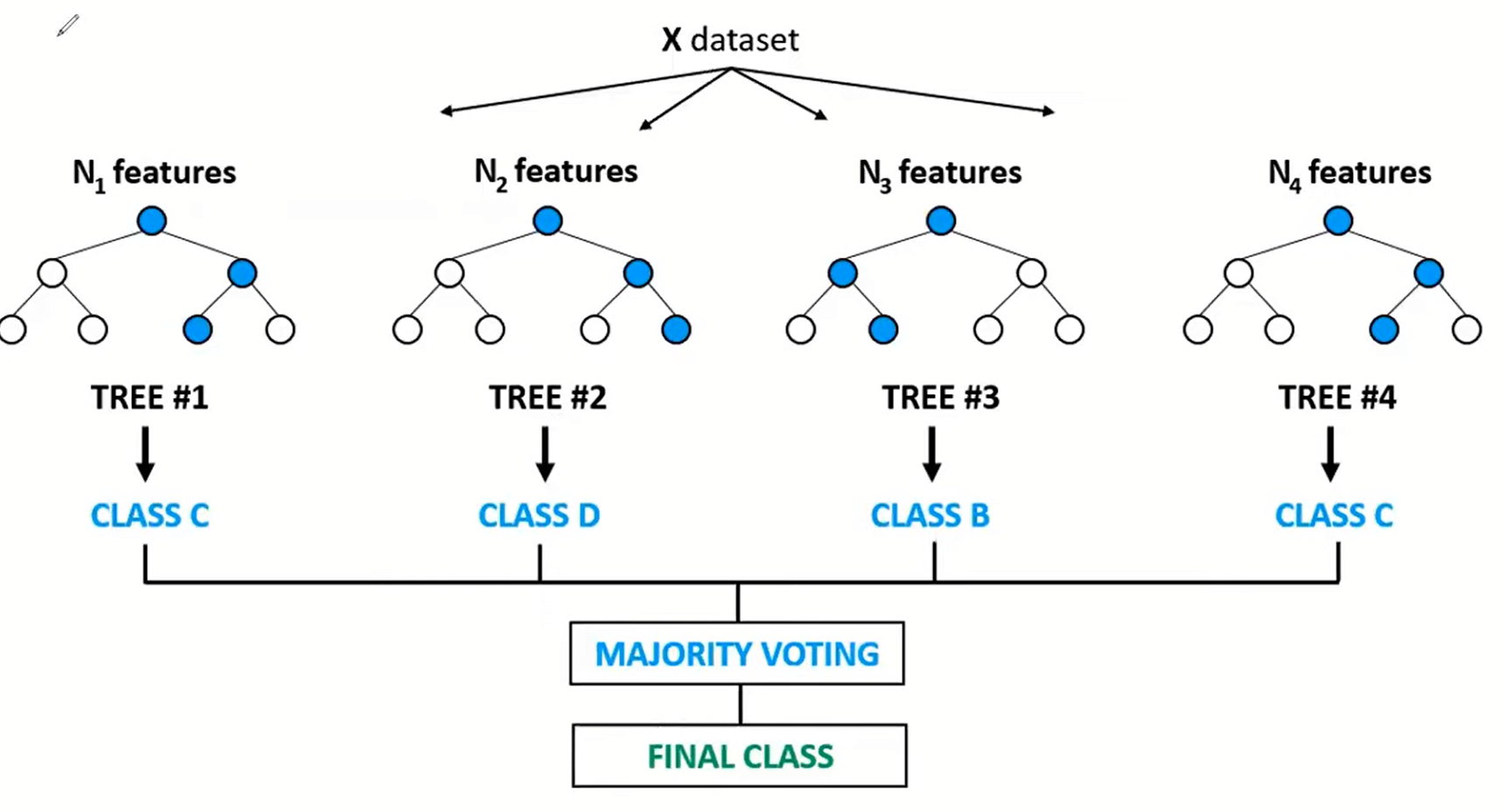
für Turnover Dataset

**Einführung**

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



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



**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

**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