**Entwurf**

* Es gibt insgesamt acht Modelle und den Dax
  + Es sind zwei Modelle im dazu gekommen, welche im ersten Teil noch nicht vorhanden waren
  + Die zwei neuen Modelle bestehen aus einer passiv-verwalteten und einem aktiv-verwalteten statistischem Modell mit modifizierten Mean-Variance-Modell. Um die statistischen Modelle korrekt mit den Modellen des maschinellen Lernens vergleichen zu können, war es wichtig diese noch hinzuzufügen.
* Diese vier statistischen Modelle wurden zudem noch doppelt ausgeführt: Einmal mit Daten für die Berechnung der Standardabweichung und der erwarteten Rendite, die die Daten der letzten 4 Jahr verarbeiten konnten und zum anderen nur mit Daten die aus dem letzte Jahr stammten.
* Für die statistischen Modelle gibt es 4 Pythonskripte
  + Ein Pythonskript für das aktiv-verwaltete statistische Modell
  + Ein Pythonskript für das passiv-verwaltete statistische Modell
  + Ein Pythonskript für das aktiv-verwaltete statistische Modell mit modifizierten Mean-Varianz-Modell
  + Ein Pythonskript für das passiv-verwaltete statistische Modell mit modifizierten Mean-Varianz-Modell
  + Generell ist nochmal anzumerken, dass aktiv-verwaltete Modelle monatlich neu gewichtet werden und eine Laufzeit von einem Jahr haben und passive Modelle nur einmal am Anfang der Jahresperiode gewichtet werden und dann das ganze Jahr mit der gleichen Gewichtung laufen
* Es gibt auch 4 Modelle mit maschinellem Lernen
  + Ein Pythonskript für das aktiv-verwaltete ML-Modell
  + Ein Pythonskript für das passiv-verwaltete ML-Modell
  + Ein Pythonskript für das aktiv-verwaltete ML-Modell mit modifiziertem Mean-Varianz-Modell
  + Ein Pythonskript für das passiv-verwaltete ML-Modell mit modifiziertem Mean-Varianz-Modell
* Wenn kein modifiziertes Mean-Varianz-Modell benutzt wurde, wurde natürlich das normale Mean-Varianz-Modell benutzt.

Wie wurde das modifizierte Portfolio erstellt?

Generell über dem Aufbau des Codes reden?

Bei allen Modellen werden am Anfang wichtige Librarys importiert.

Aufbau der statistischen Modelle:

* Zunächst wird bei den aktiv verwalteten Portfolios die .json-Datei mit dem Profit der letzten Periode eingelesen
* Beim modifizierten Modell werden die Dividenden eingelesen und auf die Rendite addiert
* Danach werden die Finanzdaten je nach Modell für das letzte Jahr bzw. für die letzten 4 Jahre heruntergeladen
  + Dabei werden der Eröffnungs-, der Höchstwert, der Tiefstwert, der Schlusskurs und das Volumen erfasst
  + Die Daten werden auf fehlende Werte geprüft
    - Falls Werte fehlen, werden diese durch den Mittelwert der vorherigen und der folgenden Zeile gebildet
  + Danach wird der letzte Tag vor der vorherzusagenden Periode ermittelt.
    - Aus diesem werden die Schlusswerte extrahiert, um den Portfoliowert am Anfang des Zyklus bewerten zu können
  + Danach werden die Finanzdaten der aktuellen Periode von Yahoo Finance heruntergeladen
    - Der Schlusswert des letzten Tages der vorherzusagenden Periode wird extrahiert, um den Wert des Portfolios am Ende des Zyklus bestimmen zu können
* Danach werden die Daten aus den letzten 4 Jahren bzw. aus dem letzten Jahr in Blöcken von 20 bzw. 252 Dateneinträgen aufgeteilt
  + Denn 20 Werktage entsprechen ungefähr einem Kalendermonat
  + 252 Werktage entsprechen einem Kalenderjahr
* Pro Block wird die Rendite ermittelt
* Aus den Renditen pro Block wird die durchschnittliche Rendite über alle Blöcke gebildet
  + Beim modifizierten Modell wird die Rendite der risikofreien Anlage hinzugefügt
* Danach wird die Standardabweichung über den gleichen Zeitraum errechnet
  + Beim modifizierten Modell wird die Standardabweichung von 0 der risikofreien Anlage hinzugefügt
* Anschließend werden die Finanzdaten des letzten Jahres geladen und es wird eine Datei mit den Schlusskursen aller betrachteten Aktien erstellt
  + Diese Daten sind für die Kovarianzmatrix wichtig, die während der Portfolioallkation zum Einsatz kommen
* Danach werden die Mean-Varianz-Optimierung und die Portfolioallokation durchgeführt
  + Als Eingabeparameter haben diese Funktionen die erwartete Rendite, die Standardabweichung und Schlusskurse des letzten Jahres aller Aktien für die Kovarianzmatrix
* Anschließend wird die vorherzusagende Periode simuliert
  + Es wird bestimmt wie viel Anteile einer Aktie bzw. der risikofreien Anlage gekauft werden müssen
  + Außerdem wird der Wert des Portfolios bestimmt, um sicherzustellen, dass auch nur die 1000€ + Profit aus den letzten Perioden verwendet wurde
* Zum Schluss werden die Aktienanteile mit dem Schlusskurs des vorherzusagenden Monats addiert und der Wert des Portfolios wird ermittelt
* Hier kann ermittelt werden, wie viel Gewinn/Verlust das Portfolio gemacht hat
* Falls das Portfolio aktiv verwaltet wird, wird das Ergebnis, also der Profit, in eine .json-Datei gespeichert

Stichpunkte überprüfen auf

* Kompabilität mit allen statistischen Modellen
* Zu Fließtext umformen lassen

Aufbau der machine-learning Modelle:

Die machine-learning Modelle sind jeweils in drei Dateien aufgeteilt. Das erste Modul erstellt und trainiert die Modelle Das zweite Modul gibt die Vorhersagen aus und das dritte Modul berechnet das optimale Portfolio nach dem Mean-Varianz-Modell.

Erstes Modul „Predict20“ oder „Predict1Year“:

* Es werden Trainingsdaten pro Aktie von dem Jahr 2000 bis zum Jahr 2018 bzw. bei 2023 eingelesen
  + Je nach dem, ob das Jahr 2019 oder 2024 vorhergesagt werden sollte
* Fehlende Werte werden, wie bei den statistischen Modulen durch den Mittelwert der vorherigen und der folgenden Zeile ersetzt
* Danach werden die Daten normalisiert
* Es wird die letzte Periode vor dem vorherzusagenden Zeitraum eingelesen
  + Auf dieser basiert dann die Vorhersage
  + Fehlende Werte wurden wie bei den Trainingsdaten behandelt
  + Diese Daten werden auch normalisiert
* Es werden Datensequenzen von 60 Datenpunkten erstellt
  + Dabei wird für jeden Eingabewert „X“ der entsprechende Sollergebniswert angehängt, welcher 20 (ein Monat für aktiv verwaltete Modelle) bzw. 252 Tage (1 Jahr für passiv verwaltete Modelle) in der Zukunft liegt
    - Dadurch lernt das Modell für einen Eingabewert den Wert 20 Tage in der Zukunft vorherzusagen
* Danach werden die Trainingsdaten in Trainings-, Test-, und Validationsdaten aufgeteilt
  + Anhand der Validationsdaten wird die Leistung des Modells überprüft und Overfitting oder Underfitting zu erkennen
  + Anhand der Testdaten wird das Modell endgültig bewertet. Überprüft Generalisierungsfähigkeit des Modells
* Danach werden die Daten umgeformt und in Tensoren umgewandelt, um diese besser in das Modell einpflegen zu können
* Danach wird das Modell und die Hyperparameter definiert
* Anschließend wird das Modell evaluiert und in einer externen Datei gespeichert
* Danach wir der Testdatensatz überprüft und geschaut, wie gut die unbekannte Periode vorhergesagt werden kann
* Danach wird eine weitere unbekannte Periode als Eingabedaten dem Modell übergeben und nach der Vorhersage mit dem wirklichen Wert verglichen
  + Um den Vergleich in einer monetären Art durchführen zu können, müssen die normalisierten Werte dafür wieder in monetäre Daten umgewandelt werden

Zweites Modul:

* Es werden die Finanzdaten für die letzte Periode vor dem vorherzusagenden Zeitraum von Yahoo Finance heruntergeladen
  + Dabei werden der Eröffnungs-, der Höchstwert, der Tiefstwert, der Schlusskurs und das Volumen erfasst
  + Die Daten werden auf fehlende Werte geprüft
    - Falls Werte fehlen, werden diese durch den Mittelwert der vorherigen und der folgenden Zeile gebildet
  + Die Daten werden normalisiert
  + Die Daten werden vorbereitet
    - Dimensionen hinzugefügt
    - Daten in Tensor umgewandelt
    - Dadurch können Daten erst in Modell eingepflegt werden
  + Wie bei den statistischen Modulen
* Die machine-learning Modelle werden importiert
* Anschließend wird die Vorhersage für den Aktienwert von dem jeweiligen Modell ermittelt und diese Vorhersagen werden wieder zu monetären Werten umgewandelt
* Nun werden die wirklichen Werte dieser Periode von Yahoo Finance heruntergeladen
* Es wird die Vorhersage jeder Aktie mit dem tatsächlichen Wert verglichen
  + Der quadratische Fehler zwischen Vorhersage und tatsächlichem Wert wird ermittelt und für alle Aktien aufsummiert

Drittes Modul:

* Download der Daten der letzten Periode vor der vorherzusagenden Periode
  + Danach wird der letzte Tag vor der vorherzusagenden Periode ermittelt.
    - Aus diesem werden die Schlusswerte extrahiert, um den Portfoliowert am Anfang des Zyklus bewerten zu können
* Die Vorhersagen werden aus der .json-Datei eingelesen
* Bei dem modifizierten Mean-Varianz-Modell wird die Dividende auf die Vorhersage addiert
* Danach wird aus der Vorhersage die erwartete Rendite in Prozent berechnet
* Bei den modifizierten Portfolios wird noch die Rendite der risikofreien Anlage hinzugefügt
* Anschließend wird bei den aktiv verwalteten Portfolios der Profit der vergangenen Periode aus der .json-Datei eingelesen
* Danach werden wieder die Schlusskurse des letzten Tages der vorherzusagenden Periode aus Yahoo Finance heruntergeladen, um das Portfolio am Ende des Monats bestimmen zu können
* Anschließend werden die Finanzdaten des letzten Jahres geladen und es wird eine Datei mit den Schlusskursen aller betrachteten Aktien erstellt
  + Diese Daten sind für die Kovarianzmatrix wichtig, die während der Portfolioallkation zum Einsatz kommen
  + Bei dem modifizierten Mean-Varianz-Modell wird die risikofreie Anlage hinzugefügt
* Danach werden die Mean-Varianz-Optimierung und die Portfolioallokation durchgeführt
  + Als Eingabeparameter haben diese Funktionen die Vorhersagen der machine-learning Modelle und die Schlusskurse des letzten Jahres aller Aktien für die Kovarianzmatrix
* Anschließend wird die vorherzusagende Periode simuliert
  + Es wird bestimmt wie viel Anteile einer Aktie bzw. der risikofreien Anlage gekauft werden müssen
  + Außerdem wird der Wert des Portfolios bestimmt, um sicherzustellen, dass auch nur die 1000€ + Profit (bei den aktiv verwalteten Portfolios) aus den letzten Perioden verwendet wurde
* Zum Schluss werden die Aktienanteile mit dem Schlusskurs des vorherzusagenden Monats addiert und der Wert des Portfolios wird ermittelt
* Hier kann ermittelt werden, wie viel Gewinn/Verlust das Portfolio gemacht hat
* Falls das Portfolio aktiv verwaltet wird, wird das Ergebnis, also der Profit, in eine .json-Datei gespeichert

**Random Search**

* Um gute Ergebnisse zu erhalten, müssen auch einige Hyperparameter eingestellt werden, wie z. B. die Anzahl der Schichten, die Anzahl der Neuronen pro Schicht oder die Lernrate
* Das kann über Ausprobieren erreicht werden oder es wird eine Methode gewählt, die die verschiedenen Hyperparametereinstellungen bewertet
* Da kommt die Random-Search-Methode ins Spiel
  + Diese ist geeignet, wenn das Modell nicht besonders komplex ist
  + Bewertet verschiedene zufällige Kombinationen von Parametern aus einem Suchraum
* Vorteile der Random-Search-Methode
  + Effizienter als z.B. Grid-Search, bei der alle Kombinationen systematisch ausprobiert werden
  + Schnelles Aufspüren guter Kombinationen, ohne den gesamten Suchraum zu durchsuchen
  + Testet nicht wie Grid-Search oft viele unnötige Kombinationen
* Nachteile:
  + Keine Garantie dafür, dass die bete Kombination gefunden wird
  + Effektivität abhängig von den durchgeführten Iterationen
    - Zu wenige Iterationen kann zu suboptimalen Ergebnissen führen
    - Zu viele Iterationen sind rechenintensiv
* Für dieses Forschungsprojekt bietet Random-Search ganz klar die meisten Vorteile
  + Denn das Modell ist nicht besonders komplex
  + Es steht keine große Rechenkapazität zur Verfügung
  + Viel Zeit kann auch nicht aufgewendet werden
* Das Random-Search-Modul ist wie die Module „Predict20“ und „Predict1Year“ aufgebaut
  + Der einzige Unterschied besteht in der Definition des Models
  + Dort wurde die Random-Search-Methode implementiert
  + Davor wurden Parameterräume für die einzelnen Hyperparameter vorgegeben
  + Nach der Definition wird das Modell trainiert und bewertet
  + Das Modell mit der besten Bewertung wird ausgegeben

https://towardsdatascience.com/grid-search-vs-random-search-vs-bayesian-optimization-2e68f57c3c46

Random Search beschreiben

Insgesamt wurden acht Modelle sowie der Dax betrachtet. Neu hinzugekommen sind zwei Modelle, die im ersten Teil der Untersuchung noch nicht vorhanden waren. Diese beiden neuen Modelle umfassen ein passiv verwaltetes und ein aktiv verwaltetes statistisches Modell, das auf einem modifizierten Mean-Variance-Modell basiert. Die Hinzufügung dieser statistischen Modelle war notwendig, um einen korrekten Vergleich mit den Modellen des maschinellen Lernens zu ermöglichen.

Die vier statistischen Modelle wurden zudem in zwei Varianten ausgeführt: Einmal mit Daten, die die letzten vier Jahre umfassen, und einmal nur mit Daten aus dem letzten Jahr. Diese Differenzierung war wichtig, um die Auswirkungen unterschiedlicher Datenzeiträume auf die Modelle zu untersuchen.

Für die Umsetzung der statistischen Modelle wurden vier Pythonskripte entwickelt:

1. Ein aktiv verwaltetes statistisches Modell mit Mean-Varianz-Portfoliooptimierung
2. Ein passiv verwaltetes statistisches Modell mit Mean-Varianz-Portfoliooptimierung
3. Ein aktiv verwaltetes statistisches Modell mit modifizierter Mean-Varianz- Portfoliooptimierung
4. Ein passiv verwaltetes statistisches Modell mit modifizierter Mean-Varianz- Portfoliooptimierung

Es ist generell anzumerken, dass aktiv verwaltete Modelle monatlich neu gewichtet werden und eine Laufzeit von einem Jahr haben. Im Gegensatz dazu werden passive Modelle nur einmal zu Beginn der Jahresperiode gewichtet und behalten diese Gewichtung für das gesamte Jahr bei.

Zusätzlich wurden vier Modelle mit maschinellem Lernen (ML) implementiert:

1. Ein aktiv verwaltetes ML-Modell mit Mean-Varianz-Portfoliooptimierung
2. Ein passiv verwaltetes ML-Modell mit Mean-Varianz-Portfoliooptimierung
3. Ein aktiv verwaltetes ML-Modell mit modifizierter Mean-Varianz-Portfoliooptimierung
4. Ein passiv verwaltetes ML-Modell mit modifizierter Mean-Varianz-Portfoliooptimierung

Wenn kein modifiziertes Mean-Variance-Modell verwendet wurde, kam selbstverständlich das normale Mean-Variance-Modell zum Einsatz.

**Statistische Portfolios**

Zunächst wird bei den aktiv verwalteten Portfolios die .json-Datei mit den Profiten der letzten Periode eingelesen. Beim modifizierten Modell werden zusätzlich die Dividenden eingelesen und auf die Rendite addiert. Anschließend werden die Finanzdaten je nach Modell entweder für das letzte Jahr oder für die letzten vier Jahre heruntergeladen. Dabei werden der Eröffnungswert, der Höchstwert, der Tiefstwert, der Schlusskurs und das Volumen erfasst. Diese Daten werden auf fehlende Werte geprüft und gegebenenfalls durch den Mittelwert der vorherigen und der folgenden Zeile ersetzt. Danach wird der letzte Tag vor der vorherzusagenden Periode ermittelt, und die Schlusswerte dieses Tages werden extrahiert, um den Portfoliowert zu Beginn des Zyklus zu bewerten.

Im nächsten Schritt werden die Finanzdaten der aktuellen Periode von Yahoo Finance heruntergeladen, und der Schlusswert des letzten Tages dieser Periode wird extrahiert, um den Wert des Portfolios am Ende des Zyklus bestimmen zu können. Die Daten aus den letzten vier Jahren beziehungsweise dem letzten Jahr werden in Blöcke von 20 bzw. 252 Dateneinträgen aufgeteilt, da 20 Werktage etwa einem Kalendermonat und 252 Werktage einem Kalenderjahr entsprechen. Pro Block wird die Rendite ermittelt, und aus diesen Renditen wird die durchschnittliche Rendite über alle Blöcke gebildet. Beim modifizierten Modell wird die Rendite der risikofreien Anlage hinzugefügt.

Daraufhin wird die Standardabweichung über den gleichen Zeitraum errechnet, wobei beim modifizierten Modell die Standardabweichung von 0 der risikofreien Anlage hinzugefügt wird. Anschließend werden die Finanzdaten des letzten Jahres geladen, und es wird eine Datei mit den Schlusskursen aller betrachteten Aktien erstellt. Diese Daten sind für die Berechnung der Kovarianzmatrix wichtig, die während der Portfolioallokation zum Einsatz kommt.

Nun werden die Mean-Varianz-Optimierung und die Portfolioallokation durchgeführt. Die Eingabeparameter für diese Funktionen umfassen die erwartete Rendite, die Standardabweichung und die Schlusskurse des letzten Jahres aller Aktien, um die Kovarianzmatrix zu erstellen. Danach wird die vorherzusagende Periode simuliert, wobei bestimmt wird, wie viele Anteile einer Aktie beziehungsweise der risikofreien Anlage gekauft werden müssen. Zudem wird der Wert des Portfolios bestimmt, um sicherzustellen, dass nur die 1000 € plus die Profite aus den letzten Perioden verwendet werden.

Zum Schluss werden die Aktienanteile mit dem Schlusskurs des vorherzusagenden Monats addiert, und der Wert des Portfolios wird ermittelt, um den Gewinn oder Verlust des Portfolios zu berechnen. Falls das Portfolio aktiv verwaltet wird, wird das Ergebnis, also der Profit, in eine .json-Datei gespeichert.

**Machine-learning Portfolios**

Erstes Modul „Predict20“ und „Predict1Year“:

Zunächst werden Trainingsdaten für jede Aktie aus dem Zeitraum von 2000 bis 2018 bzw. 2023 eingelesen, je nachdem, ob das Jahr 2019 oder 2024 vorhergesagt werden soll. Fehlende Werte werden, wie bei den statistischen Modellen, durch den Mittelwert der vorherigen und der folgenden Zeile ersetzt. Danach werden die Daten normalisiert. Die letzte Periode vor dem vorherzusagenden Zeitraum wird eingelesen und ebenfalls auf fehlende Werte überprüft und normalisiert. Anschließend werden Datensequenzen von 60 Datenpunkten erstellt. Für jeden Eingabewert „X“ wird der entsprechende Sollergebniswert, der 20 Tage (ein Monat für aktiv verwaltete Modelle) bzw. 252 Tage (ein Jahr für passiv verwaltete Modelle) in der Zukunft liegt, angehängt, sodass das Modell lernt, den Wert 20 bzw. 252 Tage in der Zukunft vorherzusagen.

Die Trainingsdaten werden in Trainings-, Test- und Validationsdaten aufgeteilt. Anhand der Validationsdaten wird die Leistung des Modells überprüft, um Overfitting oder Underfitting zu erkennen. Die Testdaten dienen der endgültigen Bewertung des Modells und überprüfen dessen Generalisierungsfähigkeit. Die Daten werden umgeformt und in Tensoren umgewandelt, um diese besser in das Modell einpflegen zu können. Danach werden das Modell und die Hyperparameter definiert, das Modell evaluiert und in einer externen Datei gespeichert. Schließlich wird der Testdatensatz überprüft, um die Vorhersagefähigkeit für eine unbekannte Periode zu bewerten. Eine weitere unbekannte Periode wird als Eingabedaten dem Modell übergeben, und die Vorhersage wird mit den tatsächlichen Werten verglichen. Um diesen Vergleich in monetären Werten durchführen zu können, müssen die normalisierten Werte wieder in monetäre Daten umgewandelt werden.

Zweites Modul „PredictShares20“ und „PredictShares1Year:

Die Finanzdaten für die letzte Periode vor dem vorherzusagenden Zeitraum werden von Yahoo Finance heruntergeladen, wobei der Eröffnungswert, der Höchstwert, der Tiefstwert, der Schlusskurs und das Volumen erfasst werden. Die Daten werden auf fehlende Werte geprüft und gegebenenfalls durch den Mittelwert der vorherigen und der folgenden Zeile ersetzt. Danach werden die Daten normalisiert und vorbereitet, indem Dimensionen hinzugefügt und die Daten in Tensoren umgewandelt werden, um sie in das Modell einpflegen zu können. Anschließend werden die machine-learning Modelle importiert und die Vorhersage für den Aktienwert von dem jeweiligen Modell ermittelt und wieder zu monetären Werten umgewandelt. Die tatsächlichen Werte dieser Periode werden ebenfalls von Yahoo Finance heruntergeladen, und die Vorhersage jeder Aktie wird mit dem tatsächlichen Wert verglichen. Der quadratische Fehler zwischen Vorhersage und tatsächlichem Wert wird ermittelt und für alle Aktien aufsummiert.

Drittes Modul „MVM\_new“, „MVM\_modified“, „MVM1Year“ und „MVM1Year\_modified“:

Zunächst werden die Daten der letzten Periode vor der vorherzusagenden Periode heruntergeladen. Der letzte Tag vor der vorherzusagenden Periode wird ermittelt, und die Schlusswerte dieses Tages werden extrahiert, um den Portfoliowert zu Beginn des Zyklus zu bewerten. Die Vorhersagen werden aus der .json-Datei eingelesen, und beim modifizierten Mean-Varianz-Modell wird die Dividende auf die Vorhersage addiert. Aus der Vorhersage wird die erwartete Rendite in Prozent berechnet, und bei den modifizierten Portfolios wird noch die Rendite der risikofreien Anlage hinzugefügt. Anschließend wird bei den aktiv verwalteten Portfolios der Profit der vergangenen Periode aus der .json-Datei eingelesen. Danach werden die Schlusskurse des letzten Tages der vorherzusagenden Periode aus Yahoo Finance heruntergeladen, um das Portfolio am Ende des Monats bestimmen zu können. Die Finanzdaten des letzten Jahres werden geladen und eine Datei mit den Schlusskursen aller betrachteten Aktien erstellt. Diese Daten sind für die Kovarianzmatrix wichtig, die während der Portfolioallokation zum Einsatz kommt. Bei dem modifizierten Mean-Varianz-Modell wird die risikofreie Anlage hinzugefügt.

Danach werden die Mean-Varianz-Optimierung und die Portfolioallokation durchgeführt. Diese Funktionen verwenden die Vorhersagen der machine-learning Modelle und die Schlusskurse des letzten Jahres aller Aktien als Eingabeparameter für die Kovarianzmatrix. Anschließend wird die vorherzusagende Periode simuliert, wobei bestimmt wird, wie viele Anteile einer Aktie bzw. der risikofreien Anlage gekauft werden müssen. Der Wert des Portfolios wird ermittelt, um sicherzustellen, dass nur die 1000 € plus Profit (bei den aktiv veralteten Portfolios) aus den letzten Perioden verwendet wurden. Zum Schluss werden die Aktienanteile mit dem Schlusskurs des vorherzusagenden Monats addiert und der Wert des Portfolios wird ermittelt. Dadurch kann, wie bei den statistischen Modellen, der Gewinn oder Verlust des Portfolios ermittelt werden. Falls das Portfolio aktiv verwaltet wird, wird der Überschuss, also der Profit, in eine .json-Datei gespeichert. Ein negativer Profit würde einen Verlust darstellen.