Strukturierte Anleitung zur Entwicklung einer keramikgefüllten UV-resinbasierten 3D-Druck-Formulierung mittels Machine Learning

1 Materialsystematik & Datenstruktur

1.1 Definition der Materialklassen

Um eine systematische Datenbank aufzubauen, werden alle Materialien und deren Kenngrößen in folgende Klassen unterteilt:

- **Keramik-Partikel (Solids Content)**
  - Typ (z. B. α-Al_2O_3, Si_3N_4, ZrO_2, ...)
  - Korngröße bzw. Korngrößenverteilung (Median, D10/D90)
  - Spezifische Oberfläche (BET; m^2/g)
- **Dispergiermittel (Dispersants)**
  - Chemischer Typ (z. B. anionisch, kationisch, nichtionisch – z. B. Polycarbonsäure-Abkömmlinge, Polyacrylate)
  - Konzentration (Gewichtsprozent relativ zum Feststoffgehalt)
  - Molmasse (falls verfügbar)
- **Verdicker (Thickener)**
  - Chemischer Typ (z. B. Cellulose-Derivate, modifizierte Polyacrylate, Silikate)
  - Anteil in Vol.-% oder Gew.-%
  - Rheologische Charakteristik (thixotrop vs. newtonsch)
- **Photoinitiator**
  - Typ (z. B. CQ-Benzoylperoxid-Systeme, α-Keto-Aminophenone, Titancyaninsalze)
  - Konzentration (Gew.-% relativ zum gesamten Resin)
- **UV-Absorber**
  - Typ (z. B. Benzotriazol-Derivate, Hydroxyphenyltriazine-Derivate)
  - Konzentration (Gew.-%)
- **Monomer/Oligomer-Verhältnis**
  - Art der Monomere (z. B. TMPTA, HDDA, PEGDA)
  - Art der Oligomere (z. B. Epoxid-Methacrylate, Polyurethan-Methacrylate)
  - Massenverhältnis (z. B. 60:40 Monomer:Oligomer)

- **Nicht-reaktive Zusätze (Non-reactive Ingredients)**
  - Weichmacher (z. B. Phthalate, Adipate)
  - Stabilisatoren (z. B. Antioxidantien, UV-Stabilisatoren)
  - Lösemittelanteile (falls verwendet)

1.2 Einheitliche Protokolle zur Datenerfassung

Für jede Kenngröße wird ein standardisiertes Mess- und Dokumentationsprotokoll definiert, um Qualität und Vergleichbarkeit zu gewährleisten:

- **Viskosität**
  - Messgerät: Rotationsviskosimeter (z. B. Brookfield) oder Rheometer (z. B. Anton Paar)
  - Temperatur: stets 20 °C
  - Scherraten: $10\,\mathrm{s}^{-1}$, $60\,\mathrm{s}^{-1}$, $100\,\mathrm{s}^{-1}$
  - Prüfgeometrie: z. B. Spaltplatte 25 mm, Spaltweite 1 mm
  - Probenvolumen und Wiederholungen: Mindestens 3 Messungen pro Rezeptur
- **Aushärtetiefe (Cure Depth)**
  - Lichtquelle: UV-LED, Wellenlänge z. B. 385 nm
  - Intensität: in $\mathrm{mW/cm^2}$ (mit Radiometer messen)
  - Expositionszeit bzw. Dosis: z. B. 5 s bei $10\,\mathrm{mW/cm^2} \rightarrow 50\,\mathrm{mJ/cm^2}$
  - Prüfaufbau: Dünner Film auf Glasplatte, Messung der Cure Depth in μm (Mittelwert aus ≥ 3 Proben)
- **Sedimentationsverhalten**
  - Probenvolumen: im Transmissionsspektrometer oder Sedimentationsturmmessung (z. B. LUMiSizer)
  - Messintervalle: 1 h, 4 h, 24 h, 7 Tage
  - Bewertung: Sedimentationsdistanz in mm oder Vol.-% des sedimentierten Feststoffs
  - Optional: Partikelgrößenverteilung vor/nach Sedimentation
- **Rheologie (thixotropes Verhalten)**
  - Fließkurve: Auf- und Abstieg der Scherraten (z. B. $0 \rightarrow 100\,\mathrm{s}^{-1}$ und zurück)
  - Aufbauverfahren: Pre-Shear bei $100\,\mathrm{s}^{-1}$ für 60 s, dann $10\,\mathrm{s}^{-1}$
  - Messung der Wiederaufbauzeit: Zeit, bis die Viskosität wieder ein bestimmtes Niveau erreicht
- **Druckqualität (optisch, mikroskopisch)**
  - Standard-Testobjekt drucken (z. B. Würfel mit definierten Dimensionen und Details)
  - Fotodokumentation bei definierter Beleuchtung (z. B. 10×, 20× Mikroskop)
  - Bewertungskriterien: Oberflächenglätte, Layer-Haftung, Detailauflösung (μm)

  - Optional: Rauheitsmessung (Profilometer)

- **Mechanische Eigenschaften nach dem Sintern**
  - Bisquekörper-Sintern: Temperaturprofil, Haltezeit, Atmosphärenkontrolle
  - Dichtebestimmung: Archimedisches Verfahren
  - Biegefestigkeit: Dreipunktbiegeversuch nach DIN/ISO
  - Härte: z. B. Vickers (HV)

Hinweis: Führe für jede Rezeptur standardisierte Reihenversuche durch, um konsistente, vergleichbare Daten zu erhalten.

1.3 Datenbank-Struktur (Beispiel Excel-Blatt)

Dein bisheriger Aufbau ist bereits sehr gut. Ergänze idealerweise noch:

- **Metadaten**
  - Batch-Nummer (Rückverfolgbarkeit)
  - Datum der Herstellung/Messung
  - Operator / Laborgerät (welches Gerät wurde verwendet)
  - Umgebungsbedingungen (Temperatur, Luftfeuchtigkeit)

- **Zielgrößen**
  - Viskosität_10 (mPa · s bei $10\,\mathrm{s}^{-1}$)
  - Viskosität_60 (mPa · s bei $60\,\mathrm{s}^{-1}$)
  - Viskosität_100 (mPa · s bei $100\,\mathrm{s}^{-1}$)
  - CureDepth_50mJ (μm bei $50\,\mathrm{mJ/cm^2}$)
  - Sedimentation_24h (mm oder Vol.-% nach 24 h)
  - ThixotropeIndex (z. B. $\frac{Viskosität\_60 nach 30\ s Auf-Ab-Test}{Viskosität\_60 im Ruhezustand}$)
  - Druckqualität_Score (Skala 1–10, basierend auf Mikroskopbeurteilung)
  - Dichte_sin ($\mathrm{g/cm^3}$)
  - Biegefestigkeit (MPa)
  - Vickers_Härte (HV)

- **Kodierung der Variablen**
  - Kategoriale Eigenschaften (Materialtyp, Dispersantentyp, . . .) mit
    * Eindeutigen Codes (z. B. `A_Al2O3_300nm` statt nur $\mathrm{Al_2O_3}$)
    * Falls erforderlich, separate Spalten für Typ und Partikelgröße

- **Datenintegrität**
  - Pflichtfelder: Keine leeren Zellen bei wichtigen Eingangsgrößen
  - Konsistenzprüfungen: Summe aller Komponenten $\approx 100\,\mathrm{Gew.\text{-}\%}$
  - Validierungstabellen für zulässige Kategorien (Lookup-Tabellen)

2 Experimentelles Design & Datenerweiterung

2.1 Ziel der Datenerweiterung

- **Breite des Zusammenspiels:** Decke das gesamte sinnvolle Spektrum der Keramiktypen, Partikelgrößen und -massenanteile ab.
- **Interaktionen:** Untersuche Wechselwirkungen zwischen Dispersant, Verdicker und Monomer/Oligomer Verhältnis.
- **Reproduzierbarkeit:** Führe bei kritischen Parametern Replikate durch, um Messunsicherheiten abschätzen zu können.

2.2 Auswahl der Parameterbereiche

1. **Solids Content**
   - Minimum & Maximum: z. B. 40 Gew.-% bis 65 Gew.-%
   - Schritte: z. B. 5 %-Inkremente; in kritischen Bereichen engere Raster
2. **Dispersantentypen & -konzentrationen**
   - Typ A (z. B. anionisch): 0,5 %, 1,0 %, 1,5 %
   - Typ B (z. B. nichtionisch): 0,75 %, 1,25 %, 1,75 %
   - ...
3. **Verdicker-Anteile**
   - Thixotroper Typ X: 0,2 %, 0,5 %, 1,0 %
   - Newtonscher Typ Y: 0,5 %, 1,0 %, 1,5 %
4. **Monomer/Oligomer-Verhältnis**
   - Prüfverhältnisse: 70:30, 60:40, 50:50, 40:60, 30:70
5. **Photoinitiator & UV-Absorber-Konzentration**
   - Photoinitiator: 1 %, 2 %, 3 %
   - UV-Absorber: 0,1 %, 0,2 %, 0,5 %
6. **Nicht-reaktive Zusätze**
   - Weichmacheranteil: 0 %, 2 %, 5 %
   - Stabilisator: 0,1 %, 0,5 %, 1 %

Hinweis: Vermeide vollständige faktoriellen Designs (sonst zu viele Versuche). Nutze stattdessen partielle faktoriellen Designs oder zentrale zusammengesetzte Designs (CCD).

2.3 Versuchsplanung (DOE-Vorschlag)

- **Screening-Phase**
  - Ziel: Grobes Verständnis, welche Faktoren dominanten Einfluss haben
  - Design: Plackett-Burman oder kleines 2-Level-Vollfaktorielles Design mit 7–8 Variablen (je zwei Stufen: Niedrig, Hoch)
  - Beispielvariablen: Solids Content, Dispersantkonzentration, Verdickeranteil, Monomer/Oligomer, Photoinitiatorkonzentration, UV-Absorber, Weichmacher
- **Optimierungs-Phase**
  - Ziel: Feinjustierung im relevanten Bereich inkl. Quadrateffekte und Interaktionen
  - Design: Central Composite Design (CCD) oder Box-Behnken-Design (für 4–5 Haupteffekte, ggf. 2 Interaktionen)
- **Replikate & Validierungsexperimente**
  - Führe Wiederholungen an Punkten durch, in denen das Modell unsicher ist (Varianzabschätzung)

2.4 Dokumentation & Qualitätskontrolle

- Erstelle ein Versuchsblatt (Template), in dem alle Parameter (Feststoffgehalt, Dispersant, Verdicker, ...) vorab eingetragen werden. Ergänze danach die gemessenen Zielgrößen.
- Notiere Messnummer und Datum/Uhrzeit jeder Messung, um Störeinflüsse nachvollziehen zu können.
- Führe nach jedem Messtag eine Daten-Konsistenzprüfung durch, um Ausreißer frühzeitig zu erkennen.

3 Datenaufbereitung (Preprocessing)

3.1 Datenbereinigung

1. **Fehlende Werte (Missing Data)**
   - Prüfen, ob Eingangsparameter (z. B. Dispersantenkonzentration) oder Zielgrößen (z. B. Viskosität) fehlen
   - Vorgehen bei fehlenden Werten:
     - Leeren Datensatz löschen (nur bei wenigen fehlenden Feldern)
     - Imputation (z. B. Mittelwert/Median, K-Nearest-Neighbor-Imputation)
     - Erneute Messung (falls möglich und kritisch)
2. **Ausreißer (Outliers)**
   - Visuelle Inspektion (Box-Plots, Scatterplots)
   - Statistische Methoden (3-Sigma-Regel, IQR-Regel)
   - Ursache prüfen: Messfehler, Dateneingabefehler oder echte Materialphänomene
   - Entweder berichtigen (bei offensichtlichem Fehler) oder markieren (bei echten Phänomenen)
3. **Konsistenz (Consistency Checks)**
   - Summe aller Anteile $\approx 100\,\text{Gew.-}\%$
   - Physikalische Plausibilität prüfen (z. B. Viskositäten innerhalb Messgrenzen, Cure Depth $<$ Filmdicke)

3.2 Feature Engineering

1. **Kodierung kategorialer Variablen**
   - *One-Hot Encoding*: Für Dispersantentyp, Verdickerklasse, Photoinitiatortyp
   - *Ordinal Encoding*: Falls Rangordnung besteht (z. B. Verdickerintensität)
   - Bei vielen Kategorien (z. B. 10 verschiedene Dispersants) kann alternative *Target-Encoding* sinnvoll sein
2. **Berechnung neuer Merkmale (Derived Features)**
   - *Füllgrad*: Aus Solids Content und Dichte der Partikel kann ein Volumenanteil abgeschätzt werden
   - *Reaktive Monomer-Fraction*: $\frac{Monomergewicht}{Monomer+Oligomer}$ (dimensionsloser Parameter)
   - *Total Viskositätsindex*: Mittlere Viskosität aus $10/60/100\,\text{s}^{-1}$ oder log-Mittelwert
   - *UV-Dosisleistung*: UV-Absorberkonzentration $\times$ Lichtintensität (relativ zu Cure Depth)
   - *Sedimentationsrate*: $\frac{Sedimentationsdistanz_{24\,\text{h}} - Sedimentationsdistanz_{4\,\text{h}}}{20\,\text{h}}$ (mm/h oder dimensionslos)
3. **Normalisierung / Skalierung**

- Kontinuierliche Variablen (Viskosität, Partikelgrößen etc.) z. B. mit Z-Score oder Min-Max-Skalierung
- *Scaler* (z. B. `StandardScaler()` in scikit-learn) nur am Trainingsdatensatz anpassen und dann auf Testdaten anwenden (keine Datenleckage)

3.3 Aufteilen in Trainings- und Testdaten

- **Train/Test Split**: 70–80 % Training, 20–30 % Test/Validierung
- Optional: Zusätzlicher Hold-Out-Datensatz (z. B. 10 % der Daten) für finale Bewertung
- **Stratifizierte Aufteilung**: Falls Zielgrößen ungleich verteilt sind (z. B. Viskosität) für Homogenität in beiden Splits

4 Auswahl und Training von Machine-Learning-Modellen

4.1 Problemdefinition

- **Regression**: Vorhersage kontinuierlicher Zielgrößen (z. B. Viskosität, Cure Depth, Sedimentationsrate)
- **Multi-Output Regression**: Mehrere Zielgrößen simultan vorhersagen (z. B. Viskosität *und* Cure Depth)
- **Multi-kriterielle Optimierung**: Später, wenn mehrere Zielgrößen gleichzeitig optimiert werden sollen (Pareto-Front)

4.2 Baseline-Modelle / einfache Algorithmen

- **Lineare Regression (Ridge/Lasso)**
  - Vorteil: Interpretierbar, schnelle Trainingszeit
  - Nachteil: Beschränkte Modellierung von Nichtlinearitäten
- **Entscheidungsbäume (Decision Tree Regressor)**
  - Vorteil: Handhabbar bei kategorialen Variablen, intuitiv
  - Nachteil: Gefahr des Overfittings
- **k-Nearest Neighbors Regression (KNN)**
  - Vorteil: Keine Parametrisierung im klassischen Sinne, einfach implementierbar
  - Nachteil: Schlechte Generalisierung bei hoher Dimensionalität

4.3 Etablierte State-of-the-Art-Algorithmen

Nach Baselines empfiehlt sich der Einsatz folgender leistungsfähiger Verfahren:

- **Random Forest Regressor**
  - Ensemble aus Entscheidungsbäumen: robust gegen Ausreißer und Nichtlinearitäten
  - Wichtige Hyperparameter:
    - Anzahl der Bäume (`n_estimators`)
    - Maximale Tiefe (`max_depth`)
    - Minimale Blattgröße (`min_samples_leaf`)
- **Gradient Boosting Regressor (XGBoost, LightGBM, `HistGradientBoostingRegressor`)**
  - Sehr leistungsfähig bei heterogenen Datensätzen
  - Hyperparameter:
    - Lernrate (`learning_rate`)
    - Anzahl der Bäume (`n_estimators`)
    - Maximale Tiefe (`max_depth`)
    - `subsample`

- **Support Vector Regression (SVR, RBF-Kernel)**
  - Leistungsfähig bei kleinen bis mittelgroßen Datensätzen, insbesondere nichtlineare Zusammenhänge
  - Anspruchsvolleres Scaling und Hyperparameter-Tuning
- **Gaussian Process Regression (GPR)**
  - Vorteil: Liefert zusätzlich Unsicherheitsabschätzung (Varianz)
  - Nachteil: Rechenintensiv ($\mathcal{O}(n^3)$), daher nur bei $n < 1000$ praktikabel
- **Künstliche Neuronale Netze (Feedforward-Netzwerke)**
  - Vorteil: Hohe Flexibilität, komplexe Zusammenhänge abbildbar
  - Nachteil: Großer Datenbedarf (500 Datensätze pro Zielgröße), langer Trainingsaufwand, aufwändiges Hyperparameter-Tuning (Anzahl Schichten, Neuronen, Aktivierungsfunktionen, Lernrate)

4.4 Hyperparameter-Optimierung

Zur systematischen Optimierung der Hyperparameter bieten sich folgende Methoden an:

- **Grid Search** (`GridSearchCV` in scikit-learn) bei überschaubarem Suchraum
- **Randomized Search** (`RandomizedSearchCV`) bei größerem Suchraum
- **Bayesian Optimization** (z. B. `scikit-optimize`, `hyperopt`, `Optuna`) für effizientes Suchen in breitem Parameterraum

Dabei ist Folgendes zu beachten:

- **Cross-Validation**
  - k-fache Kreuzvalidierung (z. B. $k = 5$) auf Trainingsdaten
  - Verhindert Overfitting, indem das Modell mehrfach an verschiedenen Teilmengen trainiert und validiert wird
- **Metriken**
  - Mittlerer quadratischer Fehler (MSE) bzw. Root-MSE (RMSE) für kontinuierliche Zielgrößen
  - Mittlerer absoluter Fehler (MAE) als robustere Alternative gegenüber Ausreißern
  - Bei Multi-Output: Durchschnitt der Fehler über alle Zielgrößen oder gewichtetes Mittel (falls einige Zielgrößen wichtiger sind)

4.5 Multi-Output-Modelle

Falls mehrere Zielgrößen gleichzeitig vorhergesagt werden sollen (z. B. Viskosität *und* Cure Depth):

- **MultiOutputRegressor** (Wrapper in scikit-learn): Ein Modell pro Ziel (z. B. je ein RandomForestRegressor)

- **Regressor Chains**: Das erste Modell sagt primäre Zielgröße voraus, dessen Vorhersage wird als zusätzliches Feature ins nächste Modell eingespeist
- **Multi-Target-Gradient Boosting**: Einige Implementierungen in LightGBM/XGBoost unterstützen mehrere Zielgrößen simultan im Boosting-Verfahren

Tipp: Wenn Zielgrößen physikalisch korreliert sind (z. B. Viskosität und Cure Depth), liefern gekoppelte Multi-Output-Modelle bessere Ergebnisse.

5 Modellbewertung & Interpretierbarkeit

5.1 Cross-Validation-Ergebnisse analysieren

- **Train/Test Split vs. Cross-Validation**:
  - Nicht nur auf den Test-Datensatz schauen, sondern mittels Cross-Validation präzise Schätzungen für Bias/Varianz erhalten
  - Dokumentation:
    * Mittlerer CV-Score (z. B. mittlerer RMSE bei k-folds)
    * Streuung (Standardabweichung) der CV-Scores

5.2 Feature-Importance

Verstehen, welche Eingangsgrößen den größten Beitrag zur Vorhersage leisten:

- **Baumbasierte Modelle (Random Forest, XGBoost)**
  - Nutzung von `model.feature_importances_`
  - Visualisierung der Top-10 Features, um z. B. Einfluss von Solids Content vs. Dispersantenanteil zu vergleichen
- **Permutation Importance**
  - Modellunabhängige Methode: Permutation eines Features im Testset, Änderung des Scores beobachten
  - Aussage über Wichtigkeit basierend auf Verschlechterung der Modellleistung
- **SHAP-Werte (SHapley Additive exPlanations)**
  - Detaillierte, lokale Erklärungen, wie ein bestimmtes Sample zu einer Vorhersage beiträgt
  - Visualisiert Interaktionen (z. B. Kombination aus hoher Feststoffkonzentration + bestimmter Dispersantentyp führt zu hohem Viskositätswert)

5.3 Residualanalyse

- Prüfe Verteilung der Residuen ($Predicted - Actual$), um systematische Abweichungen zu erkennen (z. B. Über-/Unterschätzungen bei hohen Feststoffkonzentrationen)
- Visualisiere Residuen vs. vorhergesagte Werte:
  - Liegen sie zufällig um Null?
  - Gibt es Heteroskedastizität (steigende Varianz der Residuen bei größeren Vorhersagewerten)?

6 Rezepturoptimierung (Inverse Modellierung)

6.1 Formulierung des Optimierungsproblems

Definiere klar, welche Zielgrößen maximiert bzw. minimiert werden sollen:

- **Zu maximierende Zielgrößen**
  - Solids Content (höchster Keramikanteil)
  - Cure Depth (für kurze Belichtungszeiten)
  - Mechanische Eigenschaften (Biegefestigkeit, Dichte)
- **Zu minimierende Zielgrößen**
  - Viskosität (bessere Druckbarkeit)
  - Sedimentationsrate (geringere Sedimentation)
  - Druckfehler-Score (niedriger Score = bessere Druckqualität)

Hinweis: Einige Zielgrößen stehen natürlicherweise im Konflikt (z. B. hoher Feststoffanteil → hohe Viskosität). Deshalb empfiehlt sich eine Pareto-Optimierung.

6.2 Multi-Objektive Optimierungsverfahren

- **Pareto-Front**
  - Suche Rezepte, für die keine andere Lösung in allen Zielgrößen besser ist (Pareto-optimale Punkte)
  - Darstellung als Kurve oder Fläche in mehrdimensionaler Zielgrößenlandschaft
- **Gewichtete Summe**
  - Definiere Gewichte w_i für jedes Ziel und normiere Zielgrößen $\tilde{f}_i(\mathbf{x}) \in [0, 1]$

$$J(\mathbf{x}) = w_1 \cdot \tilde{f}_1(\mathbf{x}) + w_2 \cdot \tilde{f}_2(\mathbf{x}) + \dots$$

  - Verwende klassische Optimierungsalgorithmen (z. B. Nelder–Mead, Powell, L-BFGS-B) wenn differenzierbare Zielfunktionen vorliegen
- **Bayesian Optimization (BO)**
  - Surrogatmodell (z. B. Gaussian Process) für Zielgrößen
    1. Definiere *Acquisition Function* (z. B. Expected Improvement, EI)
    2. Finde nächsten Punkt $\mathbf{x}_{next}$ durch Maximierung der Acquisition Function im zulässigen Parameterrraum
    3. Evaluierung mittels ML-Vorhersagemodell (schneller Simulator)
    4. Aktualisiere Surrogatmodell mit neuem Datenpunkt
    5. Wiederhole, bis Budget erschöpft
  - Vorteil: Reduziert Anzahl notwendiger Auswertungen im großen Parameterraum
- **Evolutionäre Algorithmen (z. B. NSGA-II)**

- Generierung einer Population möglicher Rezepturen
- Variation (Mutation, Crossover) und Selektion Pareto-optimierter Individuen
- Viele Modellabfragen nötig (läuft aber direkt mit ML-Modell zur Simulation, nicht im Labor)

6.3 Praktisches Vorgehen zur Optimierung

1. **Beschränkung der Parameterbereiche**
   - Lege sinnvolle Grenzen fest (z. B. Solids Content 40 – 60 %, Dispersantenkonzentration 0,5 – 2 %)
   - Achte auf physikalische Realisierbarkeit der Kombinationen (z. B. Monomeranteil nicht zu niedrig)
2. **Definition einer Zielfunktion (Single-Objective)**

$$\min_{\mathbf{x}} f(\mathbf{x}) = Viskosität(\mathbf{x}), \quad unterBedingung CureDepth(\mathbf{x}) \geq 200\,\mu\text{m}$$

   - Implementierung z. B. mit `scipy.optimize.minimize` und `constraints`
3. **Multi-Objective mit Bayesian Optimization**
   - Baue Surrogatmodell (Gaussian Process) für alle relevanten Zielgrößen auf
   - Nutze Python-Bibliotheken wie `GPyOpt`, `BoTorch` oder `scikit-optimize (skopt)`
   - Definiere eine Acquisition Function, die auf Hypervolume (Volumen der aktuellen Pareto-Punkte) abzielt, z. B. Expected Hypervolume Improvement (EHVI)
4. **Simulieren vs. Labortests**
   - Innerhalb der BO-Schleife nur ML-Modell als kleinen Simulator verwenden
   - Alle 10–20 Iterationen echtes Experiment durchführen, um Surrogat-Modell zu validieren (Active Learning)
5. **Pareto-Front Visualisierung (2D/3D)**
   - Projektion der Pareto-Front z. B. Viskosität vs. Cure Depth für verschiedene Solids Content
   - Hilft bei Auswahl des besten Kompromisses zwischen Zielgrößen

7 Experimentelle Validierung & Iteration

7.1 Prototyp-Herstellung

- Wähle 3–5 vielversprechende Rezepturen von der Pareto-Front aus, die unterschiedliche Kompromisse abbilden (z. B. sehr niedrige Viskosität, sehr hoher Feststoffanteil, Ausgewogenheit).
- Stelle diese Formulierungen im Labor her und messe erneut alle Zielgrößen (Viskosität, Cure Depth, Sedimentation, Druckqualität).
- Vergleiche experimentelle Werte mit den ML-Vorhersagen.

7.2 Modellanpassung (Retraining)

- Tritt bei Abweichungen ($> 10\,\%$ relativer Fehler) auf, integriere die neuen Messdaten in den Trainingsdatensatz.
- Trainiere das Modell neu oder finetune es, um Generalisierung in bisher unterrepräsentierten Bereichen zu verbessern (Active Learning).

7.3 Iterativer Zyklus

1. **Modell → Optimierung → Validierung → Retraining**
2. Wiederhole, bis:
   - Das Modelloptimale Rezept reproduzierbar akzeptable Werte liefert (z. B. Viskosität $\pm 5\,\%$ vom Soll; Cure Depth $\pm 10\,\%$).
   - Kein signifikanter Sedimentationsanstieg nach ≥ 7 Tagen.
3. Dokumentiere jede Iteration, um im Nachhinein nachvollziehen zu können, warum ein Parameterwert gewählt wurde.

8 Software-/Tool-Empfehlungen und Ressourcen

8.1 Datenverarbeitung & Visualisierung

- **Python (3.8)**
  - `pandas`: Datenrahmen, CSV/Excel-Import, -Manipulation
  - `numpy`: Numerische Operationen
  - `matplotlib`: Plots (Scatter, Residuenplots, Pareto-Front)
- **Jupyter Notebook oder JupyterLab**
  - Interaktive Dokumentation der Data-Science-Schritte

8.2 Machine Learning-Bibliotheken

- **scikit-learn**
  - Algorithmen: `RandomForestRegressor`, `SVR`, `GradientBoostingRegressor`, `MultiOutputRegressor`
  - Preprocessing: `StandardScaler`, `OneHotEncoder`
  - Cross-Validation: `GridSearchCV`, `RandomizedSearchCV`, `cross_val_score`
  - Feature Importance: `feature_importances_`, Permutation Importance
- **XGBoost / LightGBM**
  - Effiziente Implementierungen von Gradient Boosting, gut skalierbar
  - Multi-Output via `MultiOutputRegressor` möglich
- **GPyOpt / BoTorch**
  - Bayesian Optimization, insbesondere für Gaussian Process-Surrogate
  - Multi-Objective-BO: *Expected Hypervolume Improvement* (EHVI), Pareto-Front-Optimierung
- **SHAP (`shap`)**
  - Detaillierte Feature-Erklärungen, Einflussanalysen

8.3 Versuchsplanung (DOE)

- **pyDOE / pyDOE2**
  - Python-Pakete zur Erstellung von faktoriellen Designs, Taguchi-Designs, Box-Behnken, CCD
  - Generiert Design-Matrizen, die als Vorlage für Laborversuche dienen
- **Orange3 (Add-Ons für DOE)**
  - Grafische Benutzeroberfläche zur Erstellung und ersten Auswertung von DOE-Designs

8.4 Rheologie-/UV-Messtechnik

- **Rheometer-Steuerung**
  - Viele Rheometer (z. B. Anton Paar MCR-Serie) exportieren Daten direkt als CSV, sofort einlesbar in Python
- **UV-Aushärtungs-Messgeräte**
  - Geräte wie *Reichert Cure Depth Tester* oder Eigenbau-Aufbauten, die Messergebnisse als CSV liefern

8.5 Versionsverwaltung & Dokumentation

- **Git + GitHub/GitLab**
  - Skripte (Jupyter Notebooks, Python-Skripte) in Repository halten, um jeden Schritt reproduzierbar zu machen
  - Commit-Nachrichten wie Feature-Engineering: SedRate = (Sed24h – Sed4h)/20 dokumentieren
- **Elektronisches Laborjournal (ELN)**
  - Versuchsprotokolle, Batch-Nummern, Abweichungen im Protokoll erfassen (z. B. LabArchives, Benchling)

9 Zusammenfassung & Ausblick

- **Materialsystematik**
  - Klare Definition der Materialklassen und Messtools
  - Einheitliche Datenbank mit allen relevanten Metadaten und Zielgrößen (Viskosität, Aushärtetiefe, Sedimentation etc.)
- **Experimentelles Design**
  - Beginn mit Screening-Experiment (Plackett-Burman), Identifikation dominanter Parameter
  - Anschließend CCD/Box-Behnken zur Erfassung von Wechselwirkungen und Quadrateffekten
- **Datenaufbereitung**
  - Bereinigung (fehlende Werte, Ausreißer)
  - Feature-Engineering (Volumenfüllgrad, reaktive Monomer-Fraktion etc.)
  - Skalierung numerischer Variablen, Kodierung kategorialer Variablen
- **Machine Learning-Modelle**
  - Einstieg mit Baselines (lineare Regression, Entscheidungsbäume)
  - Wechsel zu Random Forest, Gradient Boosting, Gaussian Process
  - Systematische Hyperparameter-Optimierung via Cross-Validation
- **Modellbewertung & Interpretierbarkeit**
  - Feature Importance (Baum-Modelle, Permutation, SHAP)
  - Residualanalyse (Verteilung, Heteroskedastizität)
- **Rezepturoptimierung**
  - Zielfunktionen definieren (Minimiere Viskosität, maximiere Cure Depth etc.)
  - Bayesian Optimization oder evolutionäre Algorithmen zur Pareto-Front-Suche
  - Visualisierung der Pareto-Front (2D/3D) für Entscheidungsfindung
- **Validierung & Iteration**
  - Laborvalidierung ausgewählter Pareto-optimierter Rezepturen
  - Retraining des Modells bei Abweichungen, Active Learning
  - Iterativer Zyklus (Modell $\rightarrow$ Optimierung $\rightarrow$ Validierung $\rightarrow$ Retraining)
- **Dokumentation & Tools**
  - Python-Bibliotheken: `pandas`, `scikit-learn`, XGBoost, GPyOpt, SHAP
  - Versionierung (Git), elektronisches Laborjournal (ELN)
  - DOE-Pakete (pyDOE2) für Versuchsplanung

Mit diesem strukturierten Vorgehen kombinierst du systematisch Fachwissen aus Polymer- und Rheologie-Chemie mit modernen Data-Science-Techniken, um bei überschaubarem Versuchsaufwand optimal abgestimmte Resin-Formulierungen für den keramischen 3D-Druck zu identifizieren. Viel Erfolg bei der Umsetzung!