Aufgabenblatt: Vorhersage der Überlebensrate auf der Titanic mit logistischer Regression in R

1 Hintergrund:

Sie sollen mithilfe von **logistischer Regression** und anderen Klassifikationsmethoden vorhersagen, ob ein Passagier die Titanic-Katastrophe überlebt hätte. Dazu nutzen Sie den Titanic-Datensatz und vergleichen verschiedene Modellansätze.

Ihr Ziel ist es, durch **Explorative Datenanalyse (EDA)** und den Einsatz von **verschiedenen Modellen** eine möglichst präzise Vorhersage zu treffen. Anschließend testen Sie Ihr Modell mit **neuen Passagieren**.

Schritt 1: Daten einlesen & erste Analyse (EDA)

1. **Laden Sie den Titanic-Datensatz** und verschaffen Sie sich mit `glimpse()` einen Überblick.
2. **Untersuchen Sie die Verteilung der Zielvariable (`Survived`)** und analysieren Sie erste Zusammenhänge, z. B.:
   - Unterschiede nach Geschlecht (`Sex`)
   - Unterschiede nach Ticketklasse (`Pclass`)
   - Einfluss des Alters (`Age`)
3. **Überprüfen Sie fehlende Werte (`NA`)** und überlegen Sie, wie Sie damit umgehen (Bspw. Imputation).

Hinweis: Nutzen Sie `ggplot2` für Visualisierungen und `dplyr` für Datenmanipulation.

Schritt 2: Datenvorbereitung & Feature Engineering

Schritt 3: Modelle testen & evaluieren

2 Aufgaben:

1. **Trainieren Sie eine logistische Regression (`glm()`).**
2. **Testen Sie mindestens zwei weitere Modelle:**
   - **Support Vector Machine (SVM) (`kernlab`-Paket)**
   - **Random Forest (`randomForest`-Paket)**
3. **Vergleichen Sie die Modelle:**

```
# Beispiel: Vergleich von echten vs. vorhergesagten Werten mit yardstick

accuracy(data = results, truth = Survived, estimate = .pred_class)

f_meas(data = results, truth = Survived, estimate = .pred_class)
```

mit Caret:

```
# Erzeuge eine Konfusionsmatrix

conf_matrix <- confusionMatrix(data = factor(preds), reference = factor(true_labels))

conf_matrix$overall['Accuracy'] # F1-Score conf_matrix$byClass['F1']
```

4. Wählen Sie das **beste Modell** aus und begründen Sie Ihre Entscheidung.

Hinweise zur Modellwahl:

- **Logistische Regression** ist ein gutes Basismodell für binäre Klassifikation.
- **SVM** kann bei komplexeren Entscheidungsgrenzen helfen. Testen Sie verschiedene Kernelfunktionen (`linear`, `radial`).
- **Random Forest** kann nichtlineare Zusammenhänge erfassen und ist oft robuster gegenüber Ausreißern.

3 Schritt 4: Workflow mit `tidymodels` & Vorhersagen für neue Passagiere

4 Aufgaben:

1. **Erstellen Sie eine Modellpipeline mit `recipes()`, `workflows()` und `parsnip()`.**
2. **Nutzen Sie Cross-Validation (`rsample`), um die Performance der Modelle objektiv zu bewerten.**
3. **Trainieren Sie das beste Modell auf den gesamten Datensatz.**
4. **Laden Sie eine neue Passagierliste (`new_passengers.csv`)** und sagen Sie deren Überlebenswahrscheinlichkeit vorher.

Hinweise zur Modellimplementierung:

- Verwenden Sie das `tidymodels`-Framework, um Ihre Modelle sauber zu strukturieren.
- `workflows()` hilft, Vorverarbeitung (`recipes()`) und Modelltraining zu kombinieren.
- Nutzen Sie `predict()`, um auf **neue Passagiere** angewendet zu werden.

5 Schritt 5: Präsentation der Ergebnisse

Bereiten Sie eine **kurze Präsentation** mit folgenden Inhalten vor:

1. **EDA & Datenaufbereitung** – Welche Entscheidungen wurden getroffen?
2. **Vergleich der Modelle** – Welche Modelle wurden getestet und warum?
3. **Wahl des besten Modells** – Welche Ergebnisse führten zur Entscheidung?
4. **Vorhersagen für neue Passagiere** – Welche Unsicherheiten gibt es?