LU03c - Kaggle.com: Titanic Competition

Einleitung

Die Titanic-Challenge auf Kaggle ist der klassische Einstieg in die Welt der Data Science. Das Ziel ist es, basierend auf Passagierdaten vorherzusagen, ob eine Person das Unglück überlebt hat oder nicht.

Wie so oft bei solchen neuen Herausforderungen: Wo fangen ich am besten an?

Die Lösung dieser Competition ist weniger eine Frage des kompliziertesten Algorithmus, sondern vielmehr eine Frage der präzisen Datenaufbereitung. Folgen Sie dem nachfolgend beschriebenen strukturierten Vorgehen:

Leitfaden

1. Explorative Datenanalyse (EDA)

Bevor Sie mit der Modellierung beginnen, sollten Sie die zugrunde liegenden Muster verstehen.

- **Hypothesenbildung:** Untersuchen Sie den Einfluss von Merkmalen wie Geschlecht (Sex), Passagierklasse (Pclass) und Alter (Age) auf die Überlebensrate.
- **Visualisierung:** Nutzen Sie Diagramme, um Korrelationen sichtbar zu machen. Sie werden feststellen, dass das Motto „Frauen und Kinder zuerst“ in den Daten deutlich erkennbar ist.
- **Identifikation von Ausreissern:** Prüfen Sie, ob extrem hohe Ticketpreise (Fare) oder ungewöhnliche Familienkonstellationen die Daten verzerren könnten.

2. Datenbereinigung und Feature Engineering

Dieser Schritt ist entscheidend für die Qualität Ihrer Vorhersagen.

- **Umgang mit fehlenden Werten:**
  - Füllen Sie fehlende Werte im Feld Age nicht einfach mit dem globalen Durchschnitt. Nutzen Sie stattdessen den Median innerhalb der jeweiligen Anrede-Gruppen (z. B. „Master“ für Jungen, „Miss“ für junge Frauen).
  - Ersetzen Sie fehlende Werte bei Embarked durch den häufigsten Einstiegshafen.
- **Erstellung neuer Merkmale (Feature Engineering):**
  - **FamilySize:** Kombinieren Sie SibSp und Parch, um die Gesamtgrösse der Familie zu berechnen. Oft überlebten Familienmitglieder gemeinsam oder gingen gemeinsam unter.
  - **IsAlone:** Erstellen Sie eine binäre Variable, die angibt, ob ein Passagier ohne Begleitung reiste.
  - **Titel-Extraktion:** Isolieren Sie Titel wie „Dr.“, „Rev.“ oder „Lady“ aus der Namensspalte, um den sozialen Status besser abzubilden.

3. Datenvorbereitung für den Algorithmus

Maschinen lernen aus Zahlen, nicht aus Texten.

- **Encoding:** Wandeln Sie kategoriale Variablen wie das Geschlecht in ein numerisches Format um (z. B. One-Hot-Encoding für den Hafen und Label-Encoding für das Geschlecht).
- **Skalierung:** Bringen Sie numerische Werte wie das Alter und den Ticketpreis auf eine einheitliche Skala (StandardScaler), um zu verhindern, dass Variablen mit grossen Zahlenwerten das Modell dominieren.

4. Modellwahl und Validierung

Wählen Sie ein Modell, das robust gegenüber verrauschten Daten ist.

- **Modellauswahl:** Starten Sie mit einem **Random Forest Classifier**. Dieser ist exzellent darin,

nicht-lineare Beziehungen zu erkennen, und neigt weniger zu Overfitting als einfache Entscheidungsbäume.
- **Validierungsstrategie:** Verwenden Sie die k-fache Kreuzvalidierung (Cross-Validation). Teilen Sie Ihre Trainingsdaten in mehrere Teilmengen auf, um sicherzustellen, dass Ihr Modell auf unbekannten Daten stabil performt und nicht nur die Trainingsliste auswendig lernt.

5. Hyperparameter-Optimierung

Verfeinern Sie die Parameter Ihres Modells (z. B. die Anzahl der Bäume oder die maximale Tiefe im Random Forest). Werkzeuge wie GridSearchCV helfen Ihnen dabei, die optimale Konfiguration systematisch zu finden.

6. Der finale Export

Nachdem Sie Ihr Modell auf den Testdatensatz angewendet haben, erstellen Sie die Datei submission.csv. Achten Sie strikt darauf, dass die PassengerId und die Vorhersage (Survived) exakt dem geforderten Format entsprechen.

Mit diesem methodischen Vorgehen werden Sie eine Platzierung im oberen Drittel des Leaderboards erreichen, ohne auf externe Datenquellen zurückgreifen zu müssen.

Zusammenfassung des Vorgehens

Phase	Kernaktivität	Zielsetzung
Analyse	Deskriptive Statistik	Verständnis der Überlebensfaktoren
Preprocessing	Imputation & Encoding	Maschinenlesbarkeit herstellen
Engineering	Neue Variablen erschaffen	Verborgene Informationen nutzbar machen
Training	Random Forest / XGBoost	Mustererkennung und Klassifizierung
Optimierung	Hyperparameter-Tuning	Maximierung der Genauigkeit