Kurs: KI T-InfT-008 und 010 Datenmengen und Embedded Systems

Cândido Vieira

10.10.2024

Balthasar-Neumann-Technikum (BNT)

Inhaltsverzeichnis

1. Einführung in Feature Engineering

- Umwandlung von Rohdaten in nutzbare Features
- Ziel: Verbesserung der Vorhersageleistung
- Wichtig für Modellgenauigkeit

2. Bedeutung von Feature Engineering

- Einfluss auf die Modellleistung
- Zeitaufwand im ML-Prozess
- Datenaufbereitung und Transformation

3. Typen von Features

- **Numerisch**: Alter, Preis, Einkommen
- **Kategorisch**: Land
- **Zeitbasiert**: Wochentag, Uhrzeit

4. Überblick über Techniken

- **Feature Skalierung**: Normalisierung und Standardisierung
- **Diskretisierung**: Binning
- **Transformationen**: Log und Power Transformationen
- **Interaktionsfeatures**: Kombination von Variablen

5. Feature Skalierung

- **Normalisierung:** Skaliert Daten zwischen [0, 1].

- **Standardisierung:** Daten haben Mittelwert 0 und Standardabweichung 1.

5. Normalisierung vs. Standardisierung

- **Normalisierung**: Für Algorithmen wie K-Nearest Neighbors.

- **Standardisierung**: Für Modelle mit normalverteilten Daten wie lineare Regression.

6. Diskretisierung und Binning

- Kontinuierliche Daten werden in Kategorien unterteilt.

- **Beispiel:** Einteilung von Alter in Gruppen wie "Kind", "Erwachsen", "Senior".

6. Binning Methoden

- **Gleichbreiten-Binning**: Gleich große Intervalle.

- **Gleichhäufigkeits-Binning**: Gleiche Anzahl an Datenpunkten pro Bin.

7. Transformationen: Log- und Power-Transformationen

- **Log-Transformation**: Nützlich bei exponentiellen Verteilungen.

- **Power-Transformation**: Verallgemeinerung der Log-Transformation.

7. Log-Transformation Beispiel

- **Beispiel:** um Schiefe in der Verteilung zu reduzieren.

8. Interaktionsfeatures

- Kombination von zwei oder mehr Features, um komplexe Beziehungen abzubilden.

- **Beispiel:** Alter x Einkommen.

8. Polynomiale Features

- Erstellung von Potenz-Features wie x^2, x^3, um nichtlineare Beziehungen zu erfassen.

- Nützlich bei polynomialen Regressionsmodellen.

9. Text Features

- **Tokenisierung**: Aufteilen von Text in Wörter.

- **Bag-of-Words**: Repräsentiert Texte als Vektoren basierend auf Wortfrequenzen.

9. TF-IDF

- **TF-IDF:** term frequency-inverse document frequency

- Repräsentiert wichtige Wörter durch Gewichtung seltener Begriffe.

- Nützlich zur Reduzierung des Einflusses häufiger Wörter wie "und", "der".

10. Kategorische Variablen: One-Hot-Encoding

- **One-Hot-Encoding**: Wandelt eine kategoriale Variable in binäre Spalten um.

- Jede Kategorie wird zu einer eigenen Spalte mit Werten 0 oder 1.

- Beispiel: **Wochentag** -> [Montag, Dienstag, Mittwoch]

Wochentag	Montag	Dienstag	Mittwoch
Montag	1	0	0
Dienstag	0	1	0
Mittwoch	0	0	1

10. Effektkodierung

- Unterschiede zwischen Gruppen durch Vergleich der Mittelwerte.

- Alternative zu One-Hot-Encoding für lineare Modelle.

11. Feature Auswahl

- Auswahl basierend auf statistischen Tests wie Chi-Quadrat-Test.

- Nützlich zur Reduzierung der Komplexität eines Modells.

11. Feature Auswahl: Wrapper Methoden

- Modelle werden iterativ trainiert, um die besten Features zu identifizieren.

- Methoden: Vorwärts- und Rückwärtsselektion.

11. Eingebaute Methoden zur Feature Auswahl

- Modelle wie Lasso wählen Features automatisch aus.

- Nützlich zur Vermeidung von Overfitting (Überanpassung).

12. Code-Beispiel: Skalierung

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)
```

12. Code-Beispiel: One-Hot-Encoding

```
from sklearn.preprocessing import OneHotEncoder

encoder = OneHotEncoder()

X_encoded = encoder.fit_transform(X)
```

12. Feature Auswahl: Praktisches Beispiel

- Anwendung von `SelectKBest` zur Auswahl der 3 besten Features.

- Ergebnisvergleich: Vorher und Nachher.

12. Zusammenfassung

- Feature Engineering verbessert die Modellleistung erheblich.
- Wichtige Techniken: Skalierung, Diskretisierung, Interaktionsfeatures.
- Auswahl der richtigen Features spart Rechenzeit und verhindert Überanpassung.

13. Übung 1: Skalierung und One-Hot-Encoding

- Datensatz: Titanic-Daten

- Skalierung der Variablen "Alter" und "Fare".

- One-Hot-Encoding der Variablen "Sex" und "Embarked".

13. Übung 2: Diskretisierung und Binning

- Diskretisierung der Variablen "Fare".
- Gruppenbildung: Niedrig, Mittel, Hoch.
- Visualisierung der Modellleistung vor und nach der Diskretisierung.

13. Übung 3: Min-Max Scaling

- MinMaxScale der Variablen "Age".
- Min-Max Scaling: normalisiert Daten in einen Bereich von 0 bis 1.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

13. Übung 4: Min-Max Scaling

- MinMaxScale der Variablen "Age".
- Min-Max Scaling: normalisiert Daten in einen Bereich von -1 bis 1.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Referenzen

1. Brownlee, J. Data Preparation for Machine Learning, 2020, Machine Learning Mastery.
2. Kazil, J., Jarmul, K. Data Wrangling with Python, O'Reilly Media, 2016.