Maschinelles Lernen
Aufgabenblatt 03

Prof. Dr. Christoph Böhm
Hochschule München

3. Januar 2024

Aufgabe 3.1 (Logistische Regression)**.** In dieser Aufgabe sollen Sie Daten aus einem logistischen Regressionsmodell simulieren. Im Allgemeinen betrachten wir folgendes Modell:

$$P(Y = 1|X = x) = \frac{\exp(f_w(x))}{1 + \exp(f_w(x))},$$

mit Zielgröße $Y \in \{0, 1\}$, einer Einflussgröße $x \in \mathbb{R}$ und linearem Prädiktor

$$f_w(x) = w_0 + w_1 \cdot x$$

1. Simulieren Sie $n = 50$ Beobachtungen aus dem oben genannten Modell, mit $w_0 = 0.5$, $w_1 = -1.7$ und $x \sim U([-1, 1])$; wobei $U([-1, 1])$ die Gleichverteilung auf dem Intervall $[-1, 1]$ bezeichnet.
2. Visualisieren Sie die simulierten Beobachten für x und y.
3. Fitten Sie ein logistisches Regressionsmodell auf den simulierten Daten. Wie gut passen die geschätzten Parameter zu den wahren Werten? Versuchen Sie auch größere und kleiner Werte für die Stichprobengröße; z.B. $n = 15$ und $n = 100$.
4. Interpretieren Sie die Parameter des Modells $w_0 = 0.5$ und $w_1 = -1.7$.
5. Ändern Sie die Modellparameter auf $w_0 = 0.5$, $w_1 = 2$. Wie verändern sich dadurch die Daten?

Aufgabe 3.2 (Logistische Regression und Leistungsmetriken der Klassifikation)**.** In dieser Aufgabe erstellen Sie ein Klassifikationsmodell mit Hilfe logistischer Regression auf den `Weekly` Datensatz aus dem R Begleitpaket des Buches *An Introduction to Statistical Learning, with applications in R, G. James, D. Witten, T. Hastie and R. Tibshirani, Springer, 2013.* Der Datensatz enthält wöchentliche Charakteristiken des S&P 500 von 1990 bis 2010. Eine Erklärung der Features finden Sie in Tabelle 1.

1. Laden Sie die CSV `Weekly.csv` in einen Pandas DataFrame.

Feature	Bedeutung
`Year`	Jahr der Messung
`Lag1`	Prozentuale Rendite im Vergleich zur Vorwoche
`Lag2`	Prozentuale Rendite im Vergleich zu zwei Woche zuvor
`Lag3`	Prozentuale Rendite im Vergleich zu drei Woche zuvor
`Lag4`	Prozentuale Rendite im Vergleich zu vier Woche zuvor
`Lag5`	Prozentuale Rendite im Vergleich zu fünf Woche zuvor
`Volume`	Durchschnittliche tägliches Transaktionsvolumen (in Milliarden Stück)
`Today`	Rendite dieser Woche
`Direction`	Indikator, welcher angibt, ob die Woche eine positive (`Up`) oder negative (`Down`) Rendite besitzt (siehe Vorzeichen von `Today`)

Tabelle 1: Features des `Weekly` Datensatzes.

2. Verschaffen Sie sich einen Überblick über den Datensatz.

3. Erstellen Sie mit Hilfe von `sklearn.linear_model. LogisticRegression` einen Klassifikator von `Lag1` bis `Lag5` und `Volume` auf `Direction`. Teilen Sie dabei den Datensatz in zwei in etwa gleich große Trainings- und Testdatensätze bei `random_state=0` auf.

4. Erstellen Sie mit Hilfe von `sklearn.metrics.confusion_matrix` die Wahrheitsmatrix der Vorhersage auf den Testdaten. Legen Sie dabei die Reihenfolge der Labels fest auf `Up` gefolgt von `Down`.

5. Berechnen Sie manuell die Genauigkeit, die Präzision und die Trefferquote.

6. Interpretieren Sie das Ergebnis. Wenn Sie auf Kursgewinne setzen wollen, könnten Sie sich auf ihr Modell verlassen? Welche Metrik ziehen Sie für Ihre Aussage heran?

7. Trainieren Sie ein neues logistisches Regressionsmodell, diesmal lediglich von `Lag2` auf `Direction`. Verwenden Sie ebenfalls eine gleichmäßige Aufteilung des Datensatzes.

8. Erstellen Sie mit `matplotlib.pyplot` einen Plot der Klassenwahrscheinlichkeiten von `Down` und `Up` in Abhängigkeit von `Lag2` anhand des Modells für `Lag2` $\in [-50, 50]$. Sie erhalten die Klassenwahrscheinlichkeiten über `model.predict_proba`. Sie können sich eine regelmäßige X-Achse mit Hilfe von `np.arange` erzeugen lassen. Achten Sie auf eine sinnvolle Achsenbeschriftung.

9. Wo befindet sich die Entscheidungsoberfläche? Wie lautet die Klassifikationsregel des Modells?