Aufgabe 1: Lineare Regression und Gradientenabstiegsverfahren

In dieser Aufgabe erstellen Sie ein Modell mit Hilfe (eindimensionaler) linearer Regression und trainieren das Modell selbst mit Hilfe des Gradientenabstiegsverfahren. Als Trainingsdaten verwenden wir einen Datensatz mit der Anzahl der verkauften Eiskugeln bei bestimmten Außentemperaturen angelehnt an den Datensatz `https://kenandeen.wordpress.com/2015/01/21/dissecting-a-dataset/` (Stand: 2023-05-09).

1. Laden Sie die CSV `IceCream.csv` (diese finden Sie im Moodle) in einen Pandas DataFrame.
2. Erstellen Sie mit `matplotlib.pyplot.scatter` einen Scatterplot der `SoldIceCream` über die `Temperature`. Beschriften Sie die Achsen passend.
3. Beschreiben Sie kurz den Zusammenhang der beiden Features.
4. Implementieren Sie eine Funktion `train(X, Y, steps, eta)` welche den Lernalgorithmus für die eindimensionale lineare Regression mit Hilfe des Gradientenabstiegsverfahren darstellen soll.
   Die Parameter `X` und `Y` sind Listen der Einflussgöße bzw. Zielgröße, `steps` ist die Anzahl der Lernschritte und `eta` ist die Lernrate. Die Funktion soll die beiden Gewichte $\mathbf{w}_0$ und $\mathbf{w}_1$ zurückgeben.
5. Trainieren Sie ein lineares Regressionsmodell mit Hilfe dieser Funktion auf dem Feature `Temperature` und der Ausgabe `SoldIceCream` mit 200000 Schritten und Lernrate $\eta = 0.0001$. Geben Sie $\mathbf{w}_0$ und $\mathbf{w}_1$ aus.
6. Vergleichen Sie die Parameterschätzungen für den Intercept und den Koeffizienten mit dem Ergebnis der Funktion `sklearn.linear_model.LinearRegression`.
7. Was bedeuten die Gewichte konkret in diesem Fall?
8. Implementieren Sie eine Funktion `predict(x, w0, w1)` welche eine Vorhersage für die Eingabe `x` unter den Modellparametern `w0` und `w1` zurückgibt.
9. Berechnen Sie den kleinste und größte Temperatur, die in den Daten vorkommt als Variablen `xmin` und `xmax` und die entsprechenden Vorhersagen des Modells als Variablen `ymin` und `ymax`.
10. Wiederholen Sie den Scatterplot und zeichnen Sie zusätzlich via `matplotlib.pyplot.plot` die Modellvorhersage als Linie mit Hilfe der zuvor berechneten Variablen `xmin`, `xmax`, `ymin` und `ymax` ein.
11. Interpretieren Sie den Plot.

Aufgabe 2: SciKit-Learn, R^2 und mehrdimensionale Regression

Die `Boston` Housing Daten beschreiben den Median der Hauspreise ca. um 1978 in Boston (Einheit: 1000 $) abhängig von den Features wie dargestellt in Tabelle 1. Sie werden den Umgang mit SciKit-Learn üben, Modelle anhand der R^2-Statistik vergleichen und mehrdimensionale Regressionsmodelle anwenden.

Warnhinweis zum Boston Housing Datensatz: Der Boston Housing Datensatz ist im Bereich des maschinellen Lernens noch immer einer der Standard-Datensätze, um Methoden zu illustrieren. Inzwischen ist man sich aber zunehmend der vielfältigen Probleme, wie z.B. der Diskriminierung von People of Color, welche in den Daten stecken, bewusst (siehe z.B. `https://towardsdatascience.com/things-you-didnt-know-about-the-boston-housing-dataset-2e87a6f960e8` oder auch `https://fairlearn.org/main/user_guide/datasets/boston_housing_data.html`, beide Stand: 2023-05-09) und diese sollen auch hier nicht unerwähnt bleiben.

Feature	Bedeutung
CRIM	per capita crime rate by town
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.
INDUS	proportion of non-retail business acres per town.
CHAS	Charles River dummy variable (1 if tract bounds river; 0 otherwise)
NOX	nitric oxides concentration (parts per 10 million)
RM	average number of rooms per dwelling
AGE	proportion of owner-occupied units built prior to 1940
DIS	weighted distances to five Boston employment centres
RAD	index of accessibility to radial highways
TAX	full-value property-tax rate per $10,000
PTRATIO	pupil-teacher ratio by town
B	$1000(Bk-0.63)^2$ where Bk is the proportion of Black people by town
LSTAT	% lower status of the population

Tabelle 1: Features des Boston Housing Datensatzes. Beschreibung übernommen von `https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html`.

1. Verwenden Sie den gegebenen Code-Chunk, um die Daten (als CSV-Datei im Moodle verfügbar) zu laden.

```
raw_df = pd.read_csv("data/BostonHousing.csv", header = 0) # Achtung: File path ist natürl
raw_df.head()

col = "medv"
data = raw_df.loc[:, raw_df.columns != col]
target = raw_df[col]
data.head()
```

   Alternativ sind die Daten mit Hilfe von `load_boston` aus `sklearn.datasets` in eine Variable `data` verfügbar.

2. Was ist `data` und `target`?

3. Speichern Sie die Boston Feature-Daten in einem DataFrame namens `X`.
   Zeigen sie die ersten Zeilen von `X` mit Hilfe von `X.head()` an. Was stellen Sie fest?
   Ergänzen Sie ggf. die Beschriftung z.B. aus der weiter unten gegebenen Tabelle und verifizieren Sie das Ergebnis.

4. Laden Sie die Zielgröße in einen neuen DataFrame `y` und nennen Sie die Spalte `MEDV` (Median Value). Verifizieren Sie das Ergebnis mit `y.head()`.

5. Fügen Sie die beiden DataFrames `X` und `y` mit Hilfe von `pd.concat([X, y], axis = 1, sort = False)` in einem neuen DataFrame `full` zusammen und erstellen Sie eine Scatter-Matrix.
   Bei welchen Features vermuten Sie einen direkten Zusammenhang mit den Hauspreisen?

6. Erstellen Sie einen Scatterplot der Hauspreise über das Feature `LSTAT`. Achten Sie auf eine sinnvolle Achsenbeschriftung.
   Erstellen Sie die Variable `simple_model` als neues `sklearn.linear_model.LinearRegression` Objekt und trainieren Sie das lineare Regressionsmodell via `simple_model.fit` nur mit dem Feature `LSTAT` auf die Ausgabewerte `y`.
   Tipp: Mit `X[['LSTAT']]` erhalten Sie einen DataFrame, welcher nur die Spalte `LSTAT` enthält.
   Lesen Sie aus dem Modell die beiden Parameter aus.

7. Erstellen Sie eine neue Abbildung mit dem Scatterplot der Hauspreise über das Feature `LSTAT` und zeichnen Sie eine Gerade ein mit Hilfe der zuvor berechneten Modellparametern.
   Interpretieren Sie den Plot und die beiden Parameter. Glauben Sie, dass der Zusammenhang tatsächlich linear ist?

8. Berechnen Sie mit Hilfe von `simple_model.score` den R^2-Wert.
   Interpretieren Sie den R^2-Wert.

9. Wiederholen Sie die lineare Regression diesmal im Mehrdimensionalen mit den beiden Features `LSTAT` und `RM`. Interpretieren Sie die beiden Parameterschätzer. Berechnen Sie den R^2-Wert.
   Tipp: Mit `X[['LSTAT','RM']]` erhalten Sie einen DataFrame, welcher nur die Spalten `LSTAT` und `RM` enthält.

10. Fitten Sie ein neues lineares Regressionsmodell mit allen Features und berechnen Sie den R^2-Wert.

11. Trennen Sie mit Hilfe von `train_test_split` aus `sklearn.model_selection` den kompletten Datensatz zufällig in einen Trainingsdatensatz (80%) und einen Testdatensatz (20%). Setzen Sie dabei den `random_state` auf 0.

12. Was ist nun der entsprechende R^2-Wert, wenn er auf den Testdaten berechnet wird? Warum passiert die Änderung und was bedeutet sie?