Woche 9: Einführung in die Datenanalyse

In der neunten Woche tauchen Sie in die Welt der Datenanalyse und Datenvisualisierung ein. Dabei stellen Sie sich zunehmend komplexeren Datensätzen und lernen, wie Sie diese bearbeiten, analysieren und visualisieren können. In diesem Kontext werden Sie auf leistungsfähige Bibliotheken wie NumPy, Pandas und Matplotlib treffen, die in der Datenanalyse weit verbreitet sind.

NumPy hilft Ihnen dabei, mathematische und logische Operationen auf Arrays auszuführen. Pandas ist hervorragend für die Datenmanipulation und -analyse geeignet. Es bietet Datenstrukturen und Funktionen, die das Arbeiten mit strukturierten Daten erleichtern. Matplotlib ist eine Bibliothek für die Erstellung von statischen, animierten und interaktiven Visualisierungen in Python.

Sie werden lernen, wie Sie mit diesen Werkzeugen verschiedene Arten von Diagrammen erstellen und anpassen können, um Ihre Analysen zu präsentieren. Sie werden außerdem Techniken zur Datenvorverarbeitung erlernen und verstehen, wie Sie einen Datensatz für die Analyse vorbereiten. Am Ende dieser Woche werden Sie in der Lage sein, einen kompletten Datenanalyseprozess von der Datenvorbereitung bis zur Visualisierung durchzuführen.

Mit all diesen Kenntnissen sind Sie gut gerüstet, um in die Welt der Datenwissenschaft einzusteigen!

Gesamtüberblick

Hier ein Überblick über die Inhalte und Aktivitäten der aktuellen Woche:

- Selbststudium:
  - Datenanalyse
  - Daten(vor)verarbeitung
  - Datenvisualisierung
  - NumPy
  - Pandas
  - matplotlib
- Aufgabe:
  - Datensatz analysieren
  - Diagramme erstellen
  - Datenmanipulation
- Tag 10:
  - Wiederholung
  - Vertiefung:Datenanalyse
  - Vertiefung: Datenvisualisierung
  - Ergänzung: Plotly
  - Ergänzung: Dash
  - Ergänzung: Jupyter-Notebooks

Inhalte und thematische Abgrenzung

Die folgende Auflistung zeigt detailliert, welche Themen Sie in der Woche behandeln und bearbeiten. Sie sind eine Voraussetzung für die folgenden Wochen und sollten gut verstanden worden sein.

Wenn es Verständnisprobleme gibt, machen Sie sich Notizen und fragen Sie am Präsenztag nach, so dass wir gemeinsam zu Lösungen kommen können. Und denken Sie bitte immer daran: es gibt keine „dummen“ Fragen!

1. Datenanalyse:
    - Umgang mit verschiedenen Datenformaten
    - Explorative Datenanalyse
    - Implementierung von Statistiken in Python
    - Korrelationsanalyse
2. Datenvorverarbeitung
    - Techniken zur Datenbereinigung
    - Umgang mit fehlenden Daten
    - Datentransformation
    - Datennormalisierung und Skalierung
3. Datenvisualisierung:
    - Erstellen von Diagrammen und Plots mit Matplotlib
    - Interpretation von Diagrammen und Plots
4. NumPy
    - Grundlagen von NumPy
    - Erstellung und Manipulation von NumPy-Arrays
    - Durchführung von mathematischen Operationen mit NumPy
5. Pandas
    - Erstellung und Manipulation von Pandas DataFrames
    - Datenimport und -export mit Pandas
    - Grundlegende Datenanalyse mit Pandas
6. Matplotlib
    - Grundlagen von Matplotlib
    - Erstellung verschiedener Diagrammtypen
    - Anpassung von Diagrammen
    - Subplots in Matplotlib

Lernpfad

Der Lernpfad ist ein Vorschlag, in welcher Reihenfolge Sie die Inhalte der Woche angehen können. Betrachten Sie ihn gerne als eine Todo-Liste, die Sie von oben nach unten abhaken. So können Sie sicher sein, dass Sie alle wichtigen Themen bearbeitet haben und sind gut vorbereitet für die folgenden Wochen.

1. Beginnen Sie mit dem Thema Datenanalyse. Verstehen Sie, wie man mit verschiedenen Datenformaten umgeht und was explorative Datenanalyse ist. Praktizieren Sie die Implementierung von Statistiken in Python und führen Sie eine Korrelationsanalyse durch.
2. Setzen Sie Ihre Studien mit der Datenvorverarbeitung fort. Lernen Sie Techniken zur Datenbereinigung kennen und verstehen Sie, wie Sie mit fehlenden Daten umgehen können. Üben Sie Datentransformationen sowie Datennormalisierung und Skalierung.
3. Der nächste Schritt auf Ihrem Weg ist die Datenvisualisierung. Lernen Sie, wie man Diagramme und Plots mit Matplotlib erstellt und interpretiert.
4. Nun ist es an der Zeit, sich mit NumPy vertraut zu machen. Studieren Sie die Grundlagen von NumPy und lernen Sie, wie man NumPy-Arrays erstellt und manipuliert. Führen Sie mathematische Operationen mit NumPy durch.
5. Fahren Sie mit Pandas fort. Erfahren Sie, wie man Pandas DataFrames erstellt und manipuliert. Üben Sie den Datenimport und -export mit Pandas und führen Sie grundlegende Datenanalysen mit Pandas durch.
6. Schließlich wenden Sie sich Matplotlib zu. Erkunden Sie die Grundlagen von Matplotlib und lernen Sie, wie man verschiedene Diagrammtypen erstellt. Passen Sie Diagramme an und üben Sie die Erstellung von Subplots in Matplotlib.

Programmieraufgaben

Die folgenden Programmieraufgaben sollen Ihnen eine Anregung geben. Haben Sie eigene Ideen und Themen, die Sie ausprobieren wollen, dann sollten Sie diesen nachgehen. Wichtig ist vor allem, dass Sie „Dinge ausprobieren“. Und auch, dass Sie Fehler machen, sowohl syntaktische als auch semantische. Versuchen Sie diese Fehler zu finden und aufzulösen, dann gerade aus den Fehlern lernen Sie am Ende am meisten.

1. **Datenanalyse mit Pandas**: Laden Sie einen Datensatz Ihrer Wahl (z.B. von Kaggle) in einen Pandas DataFrame. Berechnen Sie einige statistische Metriken wie Durchschnitt, Median, Modus, Varianz und Standardabweichung für mindestens eine numerische Spalte.
2. **Datenvorverarbeitung mit Pandas und NumPy**: Gegeben sei ein Datensatz mit fehlenden Werten. Nutzen Sie Pandas und NumPy, um die fehlenden Werte zu identifizieren und entweder zu entfernen oder durch geeignete Werte zu ersetzen (wie z.B. den Durchschnitt oder den Median der Spalte).
3. **Datenvisualisierung mit Matplotlib**: Wählen Sie einen Datensatz und erstellen Sie mit Matplotlib verschiedene Arten von Diagrammen, um die Daten zu visualisieren. Dazu gehören z.B. ein Balkendiagramm, ein Histogramm, ein Streudiagramm und ein Boxplot.
4. **Matrixoperationen mit NumPy**: Erstellen Sie zwei 2D-Arrays mit NumPy und führen Sie verschiedene Matrixoperationen aus, darunter Addition, Subtraktion, Matrixmultiplikation, Transponierung und Berechnung des Rangs.
5. **Übergreifende Aufgabe**: Laden Sie einen Datensatz in einen Pandas DataFrame, führen Sie einige vorverarbeitende Schritte durch (z.B. das Entfernen von fehlenden Werten), berechnen Sie einige statistische Metriken und visualisieren Sie die Daten mit Matplotlib. Verwenden Sie auch NumPy, um einige zusätzliche Analysen oder Berechnungen durchzuführen.

Abschluss-Quiz

Das Quiz soll Ihnen einen ersten Hinweis auf Ihren Lernfortschritt geben. Nach unserer Einschätzung sollten Sie diese Fragen alle beantworten können, wenn Sie den Stoff der Woche durchgearbeitet und verstanden haben. Natürlich gibt es noch sehr viel mehr mögliche Fragen, dazu wollen wir auf die Literatur und das Internet verweisen. Geben Sie gerne einmal „python quizzes“ bei Google ein.

1. Welche Funktion verwenden Sie, um eine CSV-Datei in einen Pandas DataFrame zu laden?
    1. pandas.read()
    2. pandas.load_csv()
    3. pandas.read_csv()
    4. pandas.open_csv()
2. Wie erstellen Sie ein einfaches Liniendiagramm mit matplotlib?
    1. plt.plot()
    2. plt.graph()
    3. plt.line()
    4. plt.draw()
3. Was macht die Funktion pandas.DataFrame.describe()?
    1. Sie gibt den Datentyp jeder Spalte im DataFrame aus.
    2. Sie gibt eine Zusammenfassung der zentralen Tendenzen, Dispersion und Form der Verteilung eines Datensatzes aus.
    3. Sie beschreibt die Anzahl der Zeilen und Spalten im DataFrame.
    4. Sie gibt den Namen jeder Spalte im DataFrame aus.
4. Wie würden Sie fehlende Werte in einem DataFrame mit dem Durchschnitt der anderen Werte in ihrer Spalte ersetzen?
    1. df.fillna(df.mean())
    2. df.replaceNaN(df.mean())
    3. df.mean().fillna()
    4. df.replaceNull(df.average())
5. Wie führen Sie eine Element-für-Element-Multiplikation von zwei NumPy-Arrays aus?
    1. np.multiply(arr1, arr2)
    2. np.dot(arr1, arr2)
    3. arr1 * arr2
    4. Sowohl 1. als auch 3.
6. Wie wählen Sie eine Spalte aus einem DataFrame aus?
    1. df['Spaltenname']
    2. df(0)
    3. df.select('Spaltenname')
    4. df.iloc['Spaltenname']
7. Welche der folgenden Aussagen über Pandas ist falsch?
    1. Pandas-DataFrames sind im Grunde genommen zweidimensionale Arrays.
    2. Pandas ist eine Bibliothek zum Manipulieren und Analysieren von Daten.
    3. Pandas-DataFrames sind nicht veränderbar.
    4. Pandas-DataFrames können verschiedene Datentypen enthalten.
8. Was gibt die Funktion shape in NumPy aus?
    1. Die Anzahl der Elemente im Array.
    2. Die Anzahl der Dimensionen des Arrays.

    3. Die Anzahl der Zeilen und Spalten in einem Array.
    4. Den Datentyp der Elemente im Array.
9. Wie ändern Sie die Größe eines Plots in Matplotlib?
    1. plt.figure(figsize=(Breite, Höhe))
    2. plt.resize(Breite, Höhe)
    3. plt.size(Breite, Höhe)
    4. plt.dimensions(Breite, Höhe)
10. Wie berechnen Sie die Korrelation zwischen den Spalten eines DataFrame in Pandas?
    1. df.corr()
    2. df.correlation()
    3. df.compute_corr()
    4. df.calc_corr()

Antworten:

1/3 2/1 3/2 4/1 5/4 6/1 7/3 8/3 9/1 10/1

Ressourcen

Hier nun die Verweise auf Lernquellen, die uns für diese Woche und ihre Inhalte geeignet erscheinen. Je nachdem, welcher Lerntyp Sie sind, wählen Sie sich ihre bevorzugte Quelle, es ist nicht zwingend notwendig alle durchgearbeitet zu haben. Allerdings sollten die Inhalte des Lernpfads angesprochen und erstanden worden sein.

Bücher und Texte

- Pandas in Action
- Pandas in 7 Days
- Hands-on Matplotlib
- Python Data Analytics: With Pandas, NumPy, and Matplotlib

Videos

- Python Packages – Pandas
- Python Packages – numpy
- Python Packages - Matplotlib
- Work with Pandas
- Work with numpy
- Analyze Data with Pandas
- Pandas DataFrames
- Pandas Series Objects

Course

- Python & Matplotlib: Creating Box Plots, Scatter Plots, Heatmaps, & Pie Charts
- Codecademy: Learn Statistics with NumPy

Anhänge

Aktuell eine Leerseite