



Kollaboratives Programmieren in der Lehre

Ausgangssituation

Lehrveranstaltungen „BioMechatronik Praktikum“, „Sensorik“ und „Informationstechnologie im Sport“ nutzen eine Vielzahl von Software- und Hardware-Plattformen bzw. Programmiersprachen.

Herausforderungen:

- *Heterogene Technologie: Verschiedene Programmiersprachen und Tools*
- *Hoher Einrichtungsaufwand: Jede Veranstaltung erfordert 1-2 Übungstermine für die Einarbeitung*
- *Fehlende Interaktivität und Kollaboration: Studierende arbeiten für sich*
- *Fragmentierte Informationsquellen: Statische PDF-Dokumente, Code für verschiedene Tools*

Projektbeschreibung

Jupyter Notebooks als Basis für kollaboratives Lehren und Lernen.

- *Python als einheitliche Programmiersprache*
- *Einbettung von Text, Grafiken und Code in einem Dokument*
- *Interaktivität fördert den Lernerfolg (Freeman, 2014)*
- *Direkte Visualisierung erleichtert Analyse der Daten und Interpretation des Lösungswegs*
- *Können geteilt und gleichzeitig bearbeitet werden*

Maßnahmen:

- *Analyse der bestehenden Lehrmaterialien*
- *Übertragung der Aufgabenblätter und des Mustercodes*
- *Anbindung des Sensormoduls „Nicla Sense ME“*
- *Entwicklung zu einer interaktiven Lernplattform*
- *Schulung und Einweisung der Studierenden*
- *Evaluation und Feedback*

Select the 3 most interesting features (based on the results from exercise 1.3) and visualize them using fig.add_subplot(111, projection='3d') and ax.scatter() command.

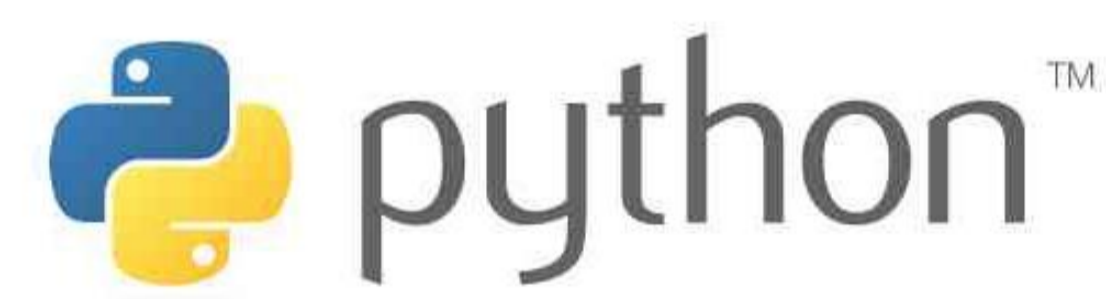
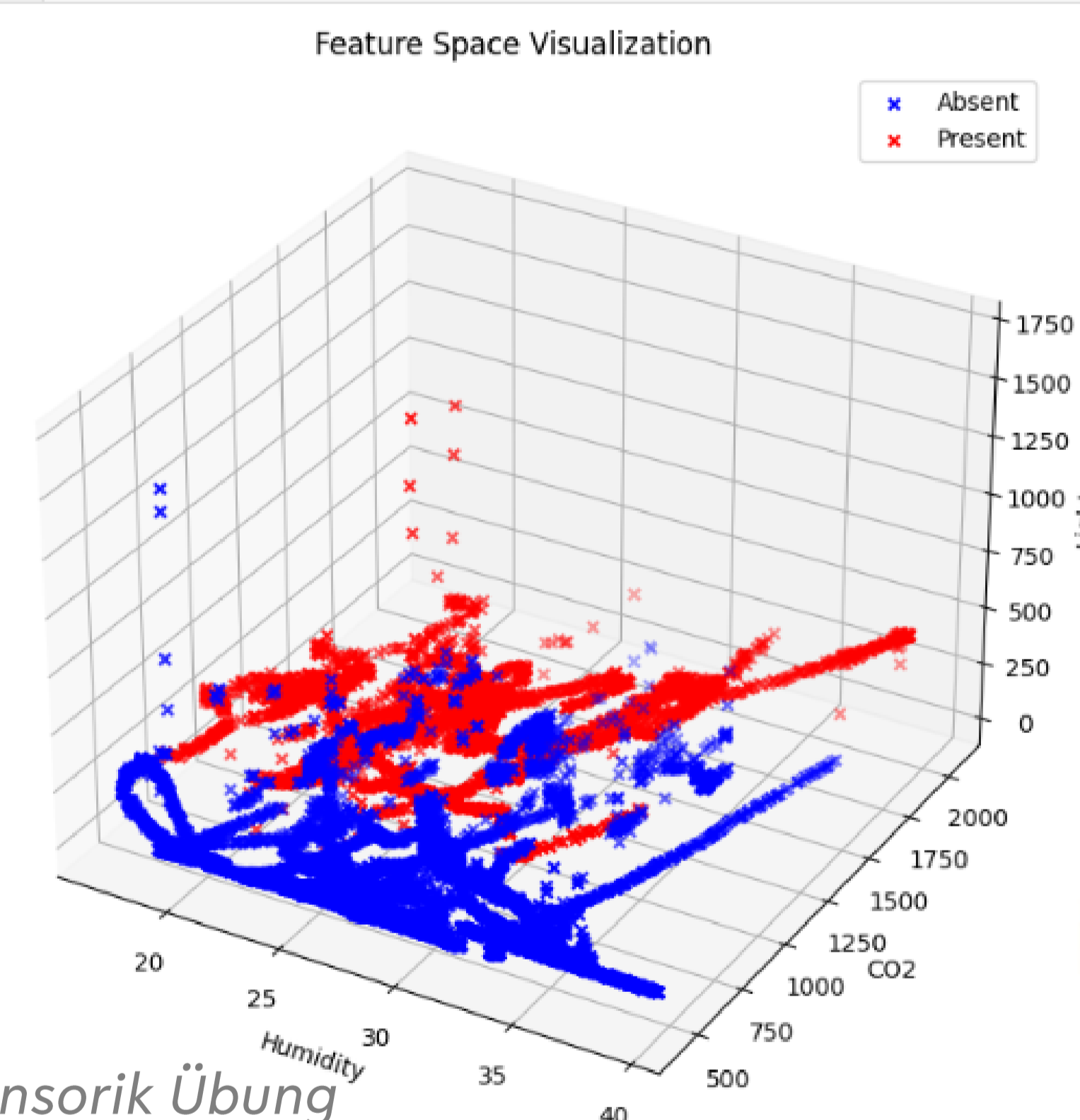
```

1 # Select the 3 most interesting features
2 selected_features = pd.DataFrame(vs_features[['CO2']])
3 selected_features['Light'] = vs_features['Light']
4 selected_features['Humidity'] = vs_features['Humidity']
5
6 # Train the decision tree model using the selected features
7 tree_selected = DecisionTreeClassifier()
8 tree_selected.fit(selected_features, vs_target)
9
10 # Predict presence or absence based on the selected features
11 presence_prediction = tree_selected.predict(selected_features)
12
13 # Plot the data points in a 3D scatter plot with colored points
14 fig = plt.figure(figsize=(10, 8))
15 fig.tight_layout()
16 ax = fig.add_subplot(111, projection='3d')
17
18 # Scatter plot for presence (1) and absence (0)
19 ax.scatter(selected_features[presence_prediction == 0][['Humidity']],
20           selected_features[presence_prediction == 0][['CO2']],
21           selected_features[presence_prediction == 0][['Light']],
22           c='b', marker='x', label='Absent')
23
24 ax.scatter(selected_features[presence_prediction == 1][['Humidity']],
25           selected_features[presence_prediction == 1][['CO2']],
26           selected_features[presence_prediction == 1][['Light']],
27           c='r', marker='x', label='Present')
28
29 ax.set_xlabel('Humidity')
30 ax.set_ylabel('CO2')
31 ax.set_zlabel('Light')
32 ax.set_title('Feature Space Visualization')
33 ax.legend()
34
35 plt.show()

```



Nicla Sense ME



Ergebnisse

Beispiel aus einer Sensorik Übung

- *Einheitliche Plattform in Hardware und Software reduziert initialen Aufwand*
- *Fokussierung auf Lerninhalte*
- *Studierende arbeiten in einer verbreiteten Entwicklungsumgebung und Programmiersprache*
- *Material kann niederschwellig aktualisiert und in den nächsten Semestern weiter verwendet werden*

Kontaktinformationen:

Dr. Marc Hesse
AG Kognitronik & Sensorik
Technische Fakultät

m.hesse@uni-bielefeld.de
0521 106 12041

Fazit

- *Förderung hat Raum für grundlegende Überarbeitung der Veranstaltungen gegeben*
- *Ansatz kann auf weitere Lehrveranstaltungen übertragen werden*
- *Etwaige Skalierung auf hochschulweite Umsetzung*