Infos zum Datensatz

- Der Datensatz liegt in Ilias komprimiert als `.zip` vor.
- Der Datensatz enthält Beschleunigungsmessungen an Kugellagern
- Für jede Messung gibt es ein Verzeichnis `Bearingx_y`, das einige CSV-Dateien `acc_#.csv` enthält, sowie eine Datei `Bearingx_y_health_state.csv` .
- In den CSV-Dateien `acc_#.csv` sind horizontale und vertikale Beschleunigung, sowie der Messzeitpunkt nach folgendem Schema erfasst:

Stunde	Minute	Sekunde	Mikrosekunde (μs)	Horiz. Beschl.	Vert. Beschl.
...	...	...	...	...	...

- In der CSV-Datei `Bearingx_y_health_state.csv` ist der Verschleißzustand für jede Datei `acc_#.csv` angegeben. Diese werden wir als Label verwenden. Die Werte stehen dabei für
  0: Lager neu
  1: Lager abgenutzt
  2: Lager stark verschlissen

Aufgaben

1. Mache dich zunächst mit dem Datensatz vertraut. Lies aus dem Ordner `Bearing1_4` die Inhalte der ersten zwei CSV-Dateien ein und plotte die horizontalen und vertikalen Beschleunigungen über den zeitlichen Verlauf. Verwende Achsenbeschriftungen und eine Legende.

   Tipp: Schreibe eine Funktion, die die einzelnen Zeitspalten zu einem *datetime*-Objekt zusammenfasst, um die Zeitdaten sinnvoll verwenden zu können. Achte außerdem auf die Separatoren in den CSV-Dateien.

2. Wie verhält sich die Vibration in den Kugellagern über den Zeitraum einer ganzen Messung? Lies dazu für Messung `Bearing1_4` nacheinander alle Beschleunigungsdateien ein. Erstelle anschließend ein Data Frame, welches die Mittelwerte und Standardabweichungen der horizontalen und vertikalen Beschleunigungen sowie das entsprechende Label aus `Bearingx_y_health_state.csv` enthält. Pro Zeile im Data Frame sollen also die Features und das Label einer CSV-Datei `acc_#.csv` enthalten sein. Plotte anschließend diese Werte über einen sinnvollen und passenden zeitlichen Verlauf. Welche Tendenzen sind zu erkennen? Untersuche auch wie die Labels zeitlich verteilt sind.

3. Erstelle nun einen Datensatz aus allen Messungen, der zum Modelltraining für eine Klassifikation verwendet werden kann. Dieser soll zunächst Mittelwerte und Standardabweichungen beider Beschleunigungen als Features, sowie den Verschleißzustand als Label enthalten.

Tipp: Es empfiehlt sich den extrahierten Datensatz mithilfe von *pandas* oder *NumPy* als Datei zu speichern, um ihn für das spätere Modelltraining direkt einlesen zu können. Achte außerdem nochmal darauf, ob der Separator in den CSV-Files immer gleich ist.

4. Im ersten Schritt möchten wir eine binäre Klassifikation durchführen. Reduziere dazu die Anzahl der Klassen von 3 auf 2 indem du die Zustände *Lager neu* und *Lager abgenutzt* zu einem Label zusammenfasst. Skaliere den Datensatz und teile ihn in Trainings- und Testdaten auf. Verwende einen Testdatenanteil von mindestens 20 %.

5. Trainiere den Datensatz auf einigen Klassifikatoren der Scikit-Learn-Bibliothek (Support Vector Machine, Random Forest und Gradient Boosting) sowie auf einem neuronalen Netz der Keras Bibliothek. Versuche geeignete Hyperparameter für die jeweiligen Verfahren zu finden. Lass dir Accuracy, Precision und Recall auf dem Testdatensatz ausgeben. Achte darauf, dass kein Overfitting stattfindet und erzeuge einen Klassifizierer, dessen Accuracy auf dem Trainingsdatensatz maximal um 1 % besser ist, als auf dem Testdatensatz. Hat dein Klassifizierer Schwächen? Falls ja, woher kommen diese?

6. Bis jetzt haben wir lediglich Mittelwert und Standardabweichung als Features aus den Rohdaten extrahiert. Die Hinzunahme weiterer Features, kann die Klassifikation erleichtern. Nutze nun auch das Maximum und Minimum jeder CSV-Datei `acc_#.csv` als weitere Features. Welche Größen fallen dir noch ein, die als Features aus den Rohdaten extrahiert werden können? Füge auch diese deinem Datensatz hinzu und führe Aufgabe 5 mit dem erweiterten Datensatz erneut aus. Können die Modelle besser werden?

7. Freiwillig: Betrachte jetzt das Klassifikationsproblem mit allen drei Klassen und trainiere erneut mit den Verfahren aus 5. Lass dir Accuracy und Konfusionsmatrix ausgeben.

Allgemeiner Hinweis
Falls die Vorverarbeitung der Daten aller neun Kugellagermessungen rechnerisch zu aufwendig für deinen Computer ist, können alle Aufgaben auch mit weniger Messungen durchgeführt werden. Allgemein empfiehlt es sich, alle Aufgaben und Vorverarbeitungsschritte zunächst nur an einem kleinen Teil des gesamten Datensatzes durchzuführen, um die korrekte Funktionalität zu testen, bevor der gesamte Datensatz genutzt wird.

Abgabe
Alle Skripte und Notebooks müssen kommentiert werden, um die Arbeitsschritte nachvollziehen zu können. Die Notebooks/Skripte müssen allesamt in einem Ordner abgelegt werden, der nach dem Studierenden benannt ist. Orientiere dich dazu in der Ordnerstruktur des Ilias. Bis **5. März 2023, 23:59 Uhr MEZ** soll der Ordner als .zip-Datei an philipp.wagner@ipa.fraunhofer.de versendet werden. Dabei müssen der Ordner `measurement_data` und andere erzeugte Datenfiles vor dem Zippen entfernt werden. Falls

es Probleme mit Paketen oder Python-Installationen gibt, melde dich bitte frühzeitig per Mail bei

philipp.wagner@ipa.fraunhofer.de oder tobias.nagel@ipa.fraunhofer.de.