Quiz – Introduction to Data Science

Dieses Übungsblatt beinhaltet Aufgaben zu dem efl-Kurs *Introduction to Data Science*, die euer Wissen zu den Kursinhalten prüfen. Bitte verwendet für die Beantwortung der Aufgaben den bereit gestellten Datensatz *winequality-red.csv*. Bitte bearbeitet diese Aufgaben eigenständig und erstellt eure Lösung in Form von dokumentiertem Code, in dem ihr eure Vorgehensweise beschreibt und auf die konkrete Frage im Quiz Bezug nehmt. Ihr könnt gerne das Code-Skeleton (*data_science_quiz_skeleton.py*) als Ausgangsbasis verwenden.

Sendet eure Lösungen bis zum **19. Januar 2025 23:59** Uhr an **dscourses@eflab.de** mit dem Betreff **„Lösungen zum Data-Science-Quiz“**. Das Abschicken eurer Lösung ist Voraussetzung für den Erhalt eines Zertifikats für die Teilnahme am oben genannten Kurs.

Zur Bearbeitung der Fragen verwendet bitte folgende Python Bibliotheken:

- os
- pandas
- numpy
- matplotlib
- seaborn
- scikit-learn

Viel Erfolg und Spaß bei den Übungen!

1. Weine in der Vorbereitung

1.1 Importiere den vorliegenden Datensatz *winequality-red.csv* mit Hilfe der *pandas* Bibliothek, so dass am Ende ein *DataFrame*-Objekt als Variable in deiner Entwicklungsumgebung vorliegt.

1.2. Führe das Pre-Processing durch, indem du den DataFrame auf NaN (Not-a-Number) Values prüfst. Sind NaN-Werte enthalten? Wenn ja, wie viele sind es pro Spalte?

1.3. Lasse dir einige deskriptive Kennzahlen über die Daten ausgeben. Welcher DataFrame-Befehl ist an dieser Stelle sinnvoll?

2. Klarer Blick trotz Wein?

2.1. Verschaffe dir einen Überblick über den Datensatz, indem du die Daten visualisierst. Nimm dir etwas Zeit und erstelle ein *pairplot* für deinen DataFrame. Was für Erkenntnisse lassen sich aus der Darstellung gewinnen? Beschreibe bitte beispielhaft die Beziehung von ein bis zwei Variablenpaaren. Zum Beispiel: „Es besteht eine lineare Beziehung zwischen x und y.“

2.2. Analysiere den Zusammenhang zwischen *quality* und *alcohol* indem du ein *Boxplot* erstellst. Dabei soll *quality* auf der x-Achse stehen. Was können wir aus dem *Boxplot* ablesen? Gibt es einen Trend hinsichtlich des Medians?

2.3 Erstelle eine Abbildung für die Verteilung der Variable *fixed acidity* mit *sns.distplot()*. Was lässt sich aus der Abbildung schließen?

2.4 Wie sehen die Korrelationen der Variablen aus? Erstelle hierzu eine *Heatmap* die auf die Spiegeldiagonale verzichtet. Passe die im Skeleton vorliegende *Heatmap* an, indem du ihr eine neue Farbe gibst, die Anzahl der Nachkommastellen auf zwei reduzierst und die Breite auf 12 Inches und die Höhe auf 6 Inches anpasst. Welche Variablenpaare weisen eine (absolute) Korrelation von größer 0,6 auf?

3. Wein im Machine Learning - Pre-Processing

3.1. Teile den Datensatz in guten und schlechten Wein ein. Wir nehmen an, dass guter Wein bei einer Wertung von 7 beginnt. Nutze zur Unterteilung die *cut*-Methode des *DataFrames*.

3.2. Erstelle einen *seaborn countplot* für die Werte von der Spalte *„quality_binary“*. Ist das Label *„quality_binary*“ gleichverteilt?

3.3. Teile nun den Datensatz in Train- und Testset. Erläutere bitte dabei die einzelnen Schritte im Code-Skeleton. Welche Auswirkungen hat bspw. die Funktion *train_test_split* und der zugehörige Parameter *test_size* sowie die Anwendung des *StandardScaler*s? Bitte erkläre allgemein, warum wir ein Train- und ein Testset verwenden.

4. Alter Wein in neuen Schläuchen: Machine Learning - Modelling and Predicting

Ziel ist es nun den Alkoholgehalt der Weine (4.1, 4.2 und 4.3) und die Qualität der Weine (4.4 ff.) zu prognostizieren. Dazu teilen wir den Datensatz zunächst in Trainings- und Testset. Anschließend verwenden wir verschiedene Machine-Learning-Modelle zur Vorhersage des Alkoholgehalts bzw. der Qualität eines Weines basierend auf verschiedenen Features.

Prognose Alkoholgehalt

4.1. Teile den Datensatz zunächst in Trainings- und Testset. Verwende dabei als Label bzw. Zielvariable *„alcohol“*. Erläutere bitte dabei die einzelnen Schritte im Code-Skeleton. Welche Auswirkungen hat bspw. die Funktion train_test_split und der zugehörige Parameter test_size sowie die Anwendung des StandardScalers? Bitte erkläre allgemein, warum wir ein Train- und ein Testset verwenden.

4.2. Bitte konfiguriere einen *Regression Tree* wie in der Vorlesung beschrieben. Du solltest auch beschreiben, warum du bestimmte Parameter wie *max_depth* gesetzt hast und welche Auswirkungen diese Parameter haben.

4.3. Berechne bzw. trainiere nun den Entscheidungsbaum zur Prognose des Alkoholgehaltes. Evaluiere anschließend seine Prognosegenauigkeit mit passenden Metriken. Plotte den Baum in einer geeigneten Darstellung mit Hilfe von *plot_tree*. Was ist das Ergebnis deines *Regression Trees*? Konnte er den Alkoholgehalt der Weine effizient vorhersagen? Wenn nicht, was könnte das Problem sein?

Prognose Weinqualität

4.4 Wiederhole die Schritte aus 4.1. Bitte verwende nun als Label bzw. Zielvariable *„quality_binary“*. Teile auch hier den Datensatz in Trainings- und Testset und skaliere mit Hilfe des StandardScalers.

4.5. Definiere nun einen *Random Forest Classifier* zur Vorhersage der (binären) Weinqualität. Was war das Ergebnis deines *Random Forest Classifiers*? Konnte er die Qualität der Weine effizient vorhersagen? Wenn nicht, was könnte das Problem sein?

4.6. Der Aufbau eines Feed-Forward neuronalen Netzes ist im Skeleton gegeben. Bitte vervollständige die fehlenden Parameter bzgl. der Modellarchitektur. Das Modell soll aus einem Input-Layer, einem Hidden-Layer mit 20 Knoten und einem Output-Layer bestehen. Als Aktivierungsfunktion der Knoten im Hidden-Layer soll die relu-Funktion verwendet werden.

4.7. Anschließend möchten wir das neurale Netz trainieren. Bitte vervollständige dazu die fehlenden Stellen im Trainingsloop. Welche Anzahl hast du für die Batch-Size und die Anzahl der Epochen festgelegt? Welchen Effekt haben die beiden Parameter auf das Training des Modells?

4.8. Evaluiere nun dein Modell auf dem Testdatensatz. Hat das neuronale Netz besser oder schlechter die Qualität der Weine vorhergesagt als der *Random Forest Classifier?* Wie erklärst du dir den Unterschied in der Modellperformance?