Institut für Telematik
Lehrstuhl für Pervasive Computing Systems
Karlsruher Institut für Technologie (KIT)
Prof. Dr.-Ing. Michael Beigl
Dr.-Ing. Till Riedel
Ployplearn Ravivanpong, Yexu Zhou, Yiran Huang, Ayse Erozan

Praktikum Smart Data Analytics

Sommersemester 2020

1. Übungsblatt

Grundlagen und Einführung in SDIL

[Abgabedeadline: 05. Mai 2020]

Bei Smart Data Analytics geht es um die Informationsgewinnung aus Daten, um Fragestellungen zu lösen und Entscheidungen zu unterstützen. In dieser Übung wiederholen Sie einige Grundlagen der Statistik und erfahren, wie die Eigenschaften eines Datensatzes die Analyse beeinflussen. Darüber hinaus werden Sie unterschiedliche Tools und Basismethoden der Datenanalytik kennen lernen.

Hinweis: Wenn eine Aufgabe keine Anwendung spezifische Werkzeuge explizit verlangt, dürfen Sie das Werkzeug ihrer Wahl benutzen.

Aufgabe 1: Vergleichbarkeit und Reproduzierbarkeit

a) Warum vergleicht man die Algorithmen, Modelle, Performance-Metriken usw.?
b) Was versteht man unter Reproduzierbarkeit bezüglich Datenanalytik?
c) Was muss man beachten, damit die Ergebnisse vergleichbar und reproduzierbar sind?
d) Geben Sie die requirement.txt Datei für Python und den Ausdruck der Sitzungsinformation von R (`sessionInfo()`) mit Ihrer Lösung zum Übungsblatt 1 ab.

Aufgabe 2: Datenexploration, deskriptive Statistik und Ausreißer

Der Datensatz Bodyfat.csv besteht aus verschiedenen Messungen des Körperumfangs sowie Schätzungen des Körperfettanteils mittels Messung der Körperdichte von 252 Männern. Der Datensatz enthält folgende Variablen:

- Density (g/cm³): Körperdichte
- Bodyfat (%): Körperfettanteil, geschätzt mittels Siris Gleichung

$$\text{bodyfat} = \frac{495}{\text{Density}} - 450$$

- Age (Jahr)
- Height (Zoll)

- Weight (Pfund)
- Umfang verschiedener Körperteile in Zentimeter

Lesen Sie den Datensatz ein und antworten Sie die folgenden Fragen.

a) Berechnen und ggf. visualisieren Sie Durchschnitt, Median, Quantil, Modus und Spannweite aller Variablen. Welche Informationen liefern diese Statistiken über die Daten?

b) Nennen Sie zwei weitere Statistiken oder Aspekte der Daten, die interessant sein könnten. Berechnen Sie beziehungsweise visualisieren diese.

c) Was sind Ausreißer? Wie können sie entstehen?

d) Welche der Statistiken in a) und b) sind robust bzw. nicht robust gegen Ausreißer? Begründen Sie.

e) Erklären Sie, inwiefern Ausreißer die Analyse beeinflussen können.

f) Wie kann man Ausreißer in Daten erkennen? Gibt es Ausreißer im Bodyfat-Datensatz? Visualisieren Sie sie, wenn möglich.

g) Wie können wir mit Ausreißern im Allgemeinen umgehen? Wie werden Sie damit im Fall des Bodyfat-Datensatzes umgehen?

h) Was kann, abgesehen von Ausreißern, die Ergebnisse, die Gültigkeit eines statistischen oder maschinell lernenden Modells und die Genauigkeit der Vorhersage beeinflussen? Nennen Sie drei Aspekte.

Aufgabe 3: Fragestellungen und Zielvariablen begrenzen die Methodenauswahl

Folgende Teilaufgaben beschreiben die Fragestellungen. Untersuchen Sie die Art der Variablen und die Verteilung der Zielvariablen. Gegebenenfalls ist eine Datenvorverarbeitung nötig. Nennen Sie die möglichen Methoden. Wählen Sie eine Methode aus und begründen Sie Ihre Auswahl. Bitte separieren Sie die Datensätze mit 80%:20% = Training:Test und führen Sie die Methode ohne Parameteroptimierung aus. Vergleichen Sie die Ergebnisse mit den vorgegebenen Methoden.

a) Der Datensatz `employment_08_09.xlsx` beinhaltet die sozioökonomischen Daten der Arbeitskräfte in den USA im April 2008 und Angaben, ob sie im April 2009 weiterhin angestellt sind. Alle Befragten waren im April 2008 angestellt. Sagen Sie basierend auf den 2008er Informationen vorher, welche Arbeitskraft 2009 arbeitslos wird. Haben ältere Arbeitskräfte ein höheres Risiko für Arbeitslosigkeit während der Finanzkrise 2008-2009?
Baseline: Häufigste Klasse, lineare Regression.

b) Der Datensatz `Growth.xlsx` besteht aus dem Wachstum (Änderung des realen Bruttoinlandsprodukts in Prozent) von 65 Ländern und Indikatoren, die theoretisch das Wachstum erklären könnten. Schätzen Sie das Wachstum der Länder im Test-Datensatz, basierend auf den Indikatoren.

Baseline: Das durchschnittliche Wachstum aller Länder.
Hinweis: Es gibt möglicherweise Dateninkonsistenz.

c) Welche Charakteristika eines weiblichen Krebses in `crabs.txt` finden männliche Krebs (sogenannte *Satellites* auf Englisch) anziehend? Schätzen Sie die Anzahl der männlichen Krebse in Test-Datensatz anhand der Charakteristika der weiblichen Krebse.
Baseline: Poisson Regression.

d) Schätzen Sie den Anteil des Rohöls im Test-Datensatz, das in Gasolin konvertiert wird, anhand anderer Indikatoren im Datensatz `gasoline.csv`.
Baseline: Beta Regression.
Hinweis: Beta Regression ist in R implementiert.

Aufgabe 4: Korrelation und Multikollinearität

a) Lesen Sie den Datensatz `Credit.csv` ein. Erkunden Sie die Daten und konvertieren Sie gegebenenfalls kategorische Variablen in ein geeignetes Format (z. B. mit `pandas.get_dummies()` oder `sklearn.model_selection.OneHotEncoder()`).

b) Separieren Sie den Datensatz in 80%-20% für Training und Testdatensatz. Was müssen Sie hier beachten?

c) Erstellen Sie aus der Variable `Ethnicity` drei binäre Variablen `Ethnicity_Caucasian`, `Ethnicity_Asian` und `Ethnicity_African`. Führen Sie eine lineare Regression mit OLS als Schätzer mit diesen Variablen und allen anderen Variablen (außer `Limit`) durch. Berechnen Sie *Root Mean Square Error* (RMSE) zwischen der Vorhersage und den wahren Werten im Test-Datensatz. Beschreiben Sie Ihre Ergebnisse.

d) Führen Sie die Analyse in 5c) erneut durch, ohne die Variable `Ethnicity_Caucasian`. Vergleichen Sie diese Ergebnisse mit denen aus 5c).

e) Führen Sie eine lineare Regression (OLS) mit folgenden Variablen durch:

   i. Income, Limit, Age

   ii. Income, Rating, Age

   iii. Income, Limit, Rating, Age

   Beschreiben Sie die Koeffizienten, Konfidenzintervalle und Signifikanzniveau. Vergleichen Sie die Ergebnisse.

f) Was ist Korrelation? Wann benutzt man Pearson Korrelation, Spearman Korrelation und Kendalls Tau? Berechnen Sie die Korrelationen zwischen allen Variablen im Trainings-Datensatz. Welche Schlussfolgerung können Sie daraus ziehen?

g) Führen Sie die Analyse in 5e) mit Random Forest, anstatt der linearen Regression durch. Variieren Sie dabei den Parameter `random_state` mit den Werten `{1, 33, 135, 123, 99, 22}` und einigen selbstgewählten Werten. Berechnen Sie MSE des Test-Datensatzes. Vergleichen Sie Feature Importance. Welche Schlussfolgerung können Sie daraus ziehen?

h) Was ist Multikollinearität? Was sind die möglichen Ursachen und Auswirkungen? Wie kann man sie erkennen? Welche Maßnahmen gibt es, wenn Variablen in den Daten kollinear sind?

Aufgabe 5: Interpretation von Modellen

a) Lesen Sie den Datensatz `Hdma.csv` ein und machen Sie sich mit den Daten vertraut.

b) Volkswirtschaftler möchten herausfinden, was die möglichen Einflussfaktoren auf die Ablehnung eines Kreditantrags sind, und ob es eine rassistische Diskriminierung gibt. Welcher Typ von Aufgabe des maschinellen Lernens ist für diese Aufgabe geeignet? Welche Methoden kann man benutzen?

c) Ist Datenvorverarbeitung nötig? Warum? Falls nötig, verarbeiten Sie die Daten vor.

d) Separieren Sie die Daten in 80%:20% = Training:Test Datensatz. Führen Sie eine logistische Regression durch. Beschreiben Sie Koeffizienten und deren Signifikanzniveau sowie die *Area Under Curve (AUC)* der *Receiver-Operator-Curve (ROC)*. Interpretieren Sie die Ergebnisse.

e) Manche Methoden in `scikit-learn` haben den Parameter `class_weight`. Wann sollte man `class_weight = 'balanced'` setzen? Ist es in dieser Aufgabe nötig?

f) Führen Sie die Klassifizierung anhand der unten genannten Methoden durch (ohne Optimierung der Parameter). Vergleichen Sie die Feature Importance von Decision Tree, Random Forest und AdaBoost mit der der logistischen Regression aus 4d). Sind Feature Importance, Koffizienten und *log* der Wahrscheinlichkeit eines Features einer Klasse (`feature_log_prob_`) direkt vergleichbar? Warum? Sagen Sie vorher, ob ein Kreditantrag im Test-Datensatz abgelehnt wird. Berechnen Sie die AUC der ROC. Beschreiben Sie Ihre Erkenntnisse.

   i. Baseline: Häufigste Klasse
   ii. Decision Tree (`criterion = 'entropy'`)
   iii. Random Forest
   iv. AdaBoost
   v. Naive Bayes (bei `scikit-learn`, bitte `ComplementNB()` benutzen)
   vi. SVM (lineare Kernel)

Aufgabe 6: Einführung in SDIL

Lesen Sie das Kapitel 11.1 (S. 77) der SDIL Documentation[1] und erstellen Sie die Anaconda Umgebung für Ihr Projekt. Lesen Sie das Kapitel 4.2 (S. 14) des SDIL Tutorials[2].

[1] http://www.sdil.de/sdil-platform-documentation.pdf
[2] http://www.sdil.de/sdil-platform-tutorials.pdf

a) Warum ist *accuracy* eine geeignete Metrik für diese Klassifikationsaufgabe?

b) Bearbeiten Sie die Aufgabe wie beschrieben. Den Beispiel-Code können Sie im SDILGit-Repository finden: `https://git.sdil.kit.edu/sdil-tutorials`. Was ist Ihre Schlussfolgerung?

c) Ist ein durchschnittliche *accuracy* eines Modells statistisch signifikant besser als das von anderen Modellen? Lesen Sie die Methodenberatung für statistische Tests der ETH Zürich (`https://www.methodenberatung.uzh.ch/de/datenanalyse_spss.html`). Wählen Sie die geeignete Methode aus und führen Sie die Analyse durch.

Aufgabe 7: Cross Validation und Grid Search

Das maschinellen Lernen „Hyperparameter-Optimierung“ beschäftigt sich mit der Auswahl einer Menge von Hyperparametern für einen Lernalgorithmus. In der Regel mit dem Ziel, ein Maß für die Leistung des Algorithmus auf einem unabhängigen Datensatz zu optimieren.

Wenn kein Expertenwissen vorhanden ist, wird klassischerweise die Gittersuche (*Grid Search*) zur Optimierung der Hyperparameter eingesetzt. Die Gittersuche ist eine einfache exhaustive Suche und eine *brute force* Suche, die auf eine manuell spezifizierte Teilmenge des Hyperparameter-Raumes eines Lernalgorithmus angewandt wird. Ein *Grid Search* Algorithmus muss von einer Performance-Metrik geleitet werden (z.B. Genauigkeit), die typischerweise durch eine Kreuzvalidierung auf dem Trainingsset gemessen wird.

Suchen Sie einen öffentlichen Datensatz[1], der für eine Klassifikationsaufgabe geeignet ist. Bitte geben Sie entweder einen Verweis auf den Datensatz an oder hängen Sie den Datensatz bei der Abgabe Ihrer Lösung an.

Im Python-Modul `sklearn.model_selection.GridSearchCV` ist ein Grid Search-Algorithmus verfügbar. Nutzen Sie diesen, um unter verschiedenen Algorithmen und Hyperparameter das „beste“ Modell zu finden. Wählen Sie dabei geeignete Zielmetrik(en) und begründen Sie Ihre Auswahl. Plotten Sie anschließend eine Kurve mit der Suchzeit, in Abhängigkeit von der Anzahl an CPU-Kernen. (`GridSearchCV` nimmt ebenso die Parallelisierungsparameter `n_jobs` an). Plotten Sie außerdem Ihre Zielmetrik in Abhängigkeit von einem Hyperparameter Ihrer Auswahl.

Aufgabe 8: Preprocessing & Pipelines

Scikit-Learn ermöglicht es mittels Pipelines, verschiedene Vorverarbeitungsschritte (Normalisierung,

Dimensionalitätsreduktion, etc.) mit einem Klassifier zu verbinden. In dieser Aufgabe beschäftigen wir uns mit dem Workflow von Datenverarbeitungsschritten mittels Pipelines.

[1] In `Öffentliche Datensatz-Repositories.pdf` (auf ILIAS) finden Sie eine Liste mit Websites. Die Liste ist nur exemplarisch und keineswegs vollständig.

a) Laden Sie den Breast Cancer Wisconsin dataset unter: https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data

   Dieses Dataset beinhaltet ein binary Label für Krebsdiagnostic (M=malignant, B=benign) und mehrere numerische Features, berechnet aus digitalisierten Bildern von Zellkernen.

b) Nutzen Sie die `sklearn.preprocessing.LabelEncoder` um die binary Label in ein numerisches Attribut zu kodieren.

c) Mit `sklearn.cross_validation.train_test_split` splitten Sie die Daten in Trainings (80%) und Testdaten (20%). Setzen Sie `random_state=1`.

d) Als Preprocessingschritte nutzen Sie `sklearn.preprocessing.StandardScaler` für feature scaling und für Dimensionalitätsreduktion ein PCA mit sklearn.decomposition. PCA und mit `n_components=2`. Als Klassifikator nutzen Sie `sklearn.linear_model.LogisticRegression` mit `random_state=1`. Packen Sie alle Schritte in eine Pipeline (`sklearn.pipeline.Pipeline`).

e) Testen Sie mittels `pipeline.score` die Genauigkeit des Modells. Sie sollten eine Genauigkeit von 0. 947 erreichen.

f) Statt PCA nutzen Sie ein Recursive Feature Elimination (RFE) zur Feature-Auswahl (`sklearn.feature_selection.RFECV`). Wie viele und welche Features sind für die Klassifikation interessant? Welche (max.) Genauigkeit auf die Testdaten kann mit diesem Schritt (statt PCA) erreicht werden?

Aufgabe 9: Predictive Maintenance

Da die Anforderungen an Systemkomplexität und Effizienz weiter steigen, hat sich die Strategie der Maschinenwartung geändert. Wo früher ein *breakdown corrective maintenance* oder ein *scheduled preventive maintenance* der Standard waren, werden jetzt intelligentere Ansätze wie Predictive Maintenance (PM) angestrebt. Im Gegensatz zu früheren Wartungsstrategien verwendet PM die historischen Zeitreihensensordaten der Maschine, um den Zustand zu bewerten. Das Ziel ist es, die Maschinen proaktiv zu warten, bevor Ausfälle auftreten, und so Ausfallzeiten zu minimieren. Ein kritischer Teil von PM ist die Schätzung der *Remaining Useful Life* (RUL). Durch die Quantifizierung der RUL bis zum Verlust der Funktionalität einer Komponente, können Ausfallzeiten und Kosten vermieden werden, indem nur Komponenten ausgetauscht werden, die bald ausfallen werden.

In dieser Aufgabe verwenden Sie den C-Mapss-Datensatz, der prognostische Benchmark-Daten zur Vorhersage der RUL ist. Weitere Informationen finden Sie im Dokument „Damage Propagation Modeling „und „Readme“. Um Ihnen das schnellere Verständnis von Daten und Aufgaben zu erleichtern, wird ein Jupyter Notebook Vorlage auf ILIAS zur Verfügung gestellt.

Die Daten wurden in Trainings- und Testsätze aufgeteilt. Das Ziel dieser Aufgabe besteht darin, bestmögliche Ergebnis mit dem Testsatz zu erzielen. Die Bewertungsmetrik ist *Root Mean Square*

Error. Sie können beliebige Modelle erstellen, z. B. maschinelles Lernen Modelle, neuronale Netzwerkmodelle oder statistische Methoden.

Bearbeiten Sie bitte diese Aufgabe mit **google Colab Jupyter Notebook**. Teilen Sie uns den Link zum Notebook, in dem der gesamte Analyseprozess und die Analyse der Ergebnisse aufgeführt sind.

Aufgabe 10: Präsentation

Fassen Sie Ihre Erkenntnisse aus dem Übungsblatt zusammen und erstellen Sie eine Präsentation. Der Vortrag darf nicht länger als 15 Minuten sein. Sie müssen nicht alle einzelnen Teilaufgaben in der ursprünglichen Reihenfolge präsentieren.
Überlegen Sie stattdessen, was für eine „Geschichte“ Sie Ihren Kommilitonen zu jeder Hauptaufgabe erzählen möchten, welche Erkenntnisse besonders interessant sind, und begleiten Sie diese mit den Ergebnissen der Teilaufgaben.

Optional: AutoML

Die Webseite der Universität Freiburg https://www.ml4aad.org/automl/ fasst das Thema AutoML auf.

a) Wählen Sie ein AutoML Package. Begründen Sie Ihre Auswahl.

b) Führen Sie die Klassifikationsaufgabe von Aufgabe 7 mit AutoML durch. Vergleichen Sie die Ergebnisse mit den Ergebnissen aus Aufgabe 7.

c) Was ist Ihre Meinung zu AutoML?