Maschinelles Lernen
Studienarbeit – Aufgabenstellung

Prof. Dr. David Spieler
Hochschule München

2. Dezember 2019

In Ihrer Studienarbeit nehmen Sie die Rolle eines Data Scientists einer Wohltätigkeitsorganisation in den USA ein. Ihre Aufgabe ist es, einen Algorithmus zu programmieren, welcher darüber entscheidet, ob Sie eine Person kontaktieren sollten und um Spenden bitten sollten oder nicht. Sie nehmen an, dass ausschlaggebend dafür ist, ob eine Person mehr oder weniger als 50.000$ Jahreneinkommen verdient.

Als Datengrundlage finden Sie im Moodle die CSV-Datei `adult.data` aus dem UCI Machine Learning Repository unter `https://archive.ics.uci.edu/ml/datasets/adult`. Jede Zeile enthält eine Reihe von Feature-Ausprägungen beschrieben in `adult.names` inklusive einer Kategorie `income`, welche die Zielvariable ist.

Aufgabenstellung

Ihre Aufgabe ist es den klassischen Workflow eines Data Scientists nachzuvollziehen. Konkret beinhaltet dies, die folgenden Aufgaben:

Aufgabe 1 (Laden der Daten) *Laden Sie die Daten aus `adult.data` in einen Pandas DataFrame.*

Aufgabe 2 (Datenaufbereitung)

1. *In den nominalen Daten sind noch unbekannte Werte gekennzeichnet durch `'?'` vorhanden. Bereinigen Sie die Daten, indem Sie alle Zeilen entfernen, die unbekannte Werte enthalten.*
2. *Entfernen Sie die Spalten `fnlwgt` und `income` als Features.*
3. *Als Target soll das Feature `income` dienen, jedoch kommt nicht jeder Algorithmus mit nominalen Features klar. Konvertieren Sie das Target daher, sodass `income` den Wert* 1 *annimmt, falls das `income` ursprünglich den Wert `'>50K'` hat und* 0 *andernfalls.*
4. *Wieviel Prozent der Personen haben ein Einkommen von mehr als 50.000$?*

5. *Was ist die Genauigkeit eines naiven Modells, welches unabhängig von den tatsächlichen Features immer weniger als 50.000$ Einkommen zuweist? Dies ist das Mindestmaß an Genauigkeit, an dem sich ihre späteren Modelle messen müssen.*

Aufgabe 3 (Transformation) *Schreiben Sie eine Methode* `transform(X)` *welche einen Feature-DataFrame* `X` *als Parameter erhält und einen transformierten DataFrame zurückgibt, bei dem*

- *Die im Wertebereich verzerrten Features* `capital_gain` *und* `capital_loss` *sollten durch Logarithmierung normalisiert werden. Verwenden Sie hierfür die Funktion*
$$f : \mathbb{R} \to \mathbb{R}, f(x) = \log(x + 1)$$
- *Anschließend sollen alle numerischen Features, d.h.* `age`, `education_num`, `capital_gain`, `capital_loss`, `hours_per_week` *auf den Wertebereich* $[0, 1]$ *normalisiert werden.*
- *Transformieren Sie alle nominalen Features via* one-hot-encoding.
- *Transformieren Sie ihre Features mit Hilfe der Methode* `transform`.
- *Splitten Sie den Datensatz in einen Trainings- und Testdatensatz, wobei der Testdatensatz eine realtive Größe 20% haben soll. Verwenden Sie für die Reproduzierbarkeit einen* `random_state=0`.

Aufgabe 4 (Training) *Wählen Sie* ***drei*** *verschiedene, in der Vorlesung behandelte Modelltypen und trainieren Sie die entsprechenden Modelle mit den Standardparameter (keine Parameter). Geben Sie zu jedem Modell die Genauigkeit auf dem Testdatensatz aus.*

Aufgabe 5 (Modellauswahl) *Wählen Sie das im vorherigen Schritt beste Modell und tunen Sie die Hyperparameter, um möglichst eine noch bessere Performance zu bekommen.*

Aufgabe 6 (Inferenz) *Sie wollen nun Ihr im letzten Schritt final gelerntes Modell anwenden, um herauszufinden, ob der Algorithmus eine konkrete Person als potentiellen Spender auswählen würde.*

- *Erstellen Sie hierfür einen DataFrame mit Werten für eine erfundene Person.*
- *Transformieren Sie diese Personen ebenfalls mit Hilfe der Methode* `transform`. *Da die Normierung nur auf größeren Datensätzen Sinn macht, vereinen Sie den ursprünglichen DataFrame und die neue Person und transformieren das Gesamtpaket.*
- *Machen Sie mit Hilfe des Modells eine Vorhersage. Würde die Person als potentieller Spender ausgewählt werden?*

Hinweise

- Die Datei `adult.data` besitzt keinen Header und die Feature-Ausprägungen enthalten überflüssige Leerzeichen. Machen Sie sich mit den entsprechenden Optionen in `Pandas.read_csv` vertraut.
- Bindestriche `'-'` in Feature-Namen sollten vermieden werden. Besser sind Unterstriche `'_'`, da auf solche Spaltenbezeichner direkt zugegriffen werden kann.
- Sie können Zeilen in einem Pandas DataFrame `df` filtern mit Hilfe von `df[(df.A OP x) & (df.B OP y) & ...]` wobei `OP` ein beliebiger binärer Operator wir z.B. `==` ist.
- Der Wertebereich einer Spalte `'A'` eines DataFrames `df` kann mit `df['A'].apply` angepasst werden.
- Verwenden Sie `sklearn.preprocessing.MinMaxScaler` für die Skalierung von Features.
- Verwenden Sie `pandas.get_dummies` für one-hot-encoding.
- Verwenden Sie `sklearn.cross_validation.train_test_split` für das Trennen von Datensätzen in Trainings- und Testdaten.
- Vereinen Sie zwei DataFrames mit Hilfe von `pd.concat((df_A, df_B))`.

Abgabe

Erstellen Sie ein ZIP-Archiv mit dem Inhalt:

1. `ML_SA_Matrikelnummer.ipynb`: Ihre Lösung der Aufgaben der Studienarbeit in Form eines Jupyter Notebooks.
2. `ML_SA_Matrikelnummer.pdf`: Ein PDF des HTML-Export des obigen Notebooks (z.B. Export nach HTML in Jupyter, Druckenfunktion Ihres Browsers in eine PDF).

und reichen Sie über Moodle ein.