Probabilistyczne Uczenie Maszynowe

Projekt 1

20 marca 2020r.

1 Przed przystąpieniem do realizacji projektu

a. ustalić skład grup projektowych
   - grupy 2-osobowe

b. wybrać zbiór danych, który będzie służył do analiz w grupie
   - https://archive.ics.uci.edu/ml/datasets.php
   - kryteria wyboru zbioru danych:
     - minimalna liczba cech: 8
     - minimalna liczba instancji: 300

c. grupy ustalają, uzupełniają w tabeli i wysyłają do akceptacji:
   - nazwę wybranego zbioru danych, na którym będą pracowali (oraz link do tego zbioru),
   - krótki opis zadania/problemu, który będzie rozważany w raporcie - można sugerować się "Default Task" z opisu zbioru danych, ale wskazana jest kreatywność,
   - wybrany sposób ewaluacji modeli i porównania ich wyników,
   - mail o tytule: `[PUMA][P1-ZGLOSZENIE] <nazwisko_1> <nazwisko_2>`

2 Realizacja projektu

W ramach projektu do zrealizowania jest część programistyczno-analityczna oraz dokumentacyjna. Wszystkie wykonane zadania programistyczno-analityczne muszą być opisane w jednoznaczny sposób w części dokumentacyjnej.

2.1 Przeprowadzenie eksploracyjnej analizy danych

Zastosować wybrane z poniższych propozycji, który jest adekwatny do zbioru i problemu, uzasadnić:

- opis zmiennych (typ, zakres wartości/rozkład, opis słowny),
- zależności zmiennych parami (correlation, scatter plots, box plots, violin plots, mosaic plots, heat maps),

- obserwacje odstające,
- preprocessing (normalizacja, selekcja cech, ekstrakcja cech, zmiana wymiarowości),

2.2 Implementacja modeli

Zastosować wybór z poniższych propozycji, który jest adekwatny do zbioru i problemu, uzasadnić; **(należy wykonać min. po 1 z pkt. a i b)**

a. proste modele probabilistyczne:
   - Naive Bayes
   - Bayesian Linear Models
   - Gaussian Process + Mixture Models
   - Inne nieparametryczne modele Bayesowskie

b. graficzne modele probabilistyczne:
   - Directed Models:
     (1) Bayesian Network
     (2) Dynamic Bayesian Network
   - Undirected Models:
     (1) Markov Random Field
     (2) Conditional Random Field
   - Restricted Boltzmann Machine
   - Deep Belief Networks

2.3 Przeprowadzenie badania zaimplementowanych modeli na wybranym zbiorze danych

a. ustalenie scenariusza eksperymentalnego (implementacja i opis procedur uczenia i przetwarzania danych, np. 15% danych zachowane do testowania modeli bądź cross-validation itd.)

b. wybór metryk do ewaluacji modeli

c. przegląd hiperparametrów modeli (rozsądny)

d. wizualizacja wyników

e. wnioski

2.4 Raport z realizacji projektu (pkt. 1 – 3)

a. raport w postaci PDF'a, preferowane narzędzie: LaTeX, język: polski bądź angielski

b. raport wraz całościowy kodem musi być umieszczony na grupowym repozytorium projektu do godziny 22 dnia poprzedzającego termin oddania projektu

c. rozdziały raportu (wraz z punktacją do oceny):

   i. **Eksploracyjna analiza danych** (4 pkt.)

   ii. **Modele** (4 pkt.)

   iii. **Eksperymenty**

   - zastosowanie prostych modeli probabilistycznych (5 pkt)
   - zastosowanie graficznego modelu probabilistycznego (6 pkt)
   - porównanie wyników wybranych modeli* (4 pkt)
     - dla klasyfikacji - miary typu f1 score, auc, precision oraz recall (przy stosowaniu istniejących implementacji wziąć pod uwagę implementację typu "micro") z wartości oczekiwanej wielu realizacji modelu
     - dla regresji - RMSE, SSE, $R\hat{2}$, z wartości oczekiwanej wielu realizacji modelu
     - dla klasteryzacji - silhouette coefficient, rand index, jaccard index, f-measure, z wartości oczekiwanej wielu realizacji modelu
     - dla jakości modeli czysto generatywnych - negative log likelihood

   *Przeprowadzić tam gdzie jest to możliwe analizę **credible interval** na poziomie ufności 0.9.

2.5 Kod i reprodukowalność eksperymentów (2 pkt.)

Efektem tego projektu powinien być również kod źródłowy zaimplementowanych modeli probabilistycznych oraz przeprowadzonych eksperymentów. Należy zadbać o jakość i czytelność tego kodu.

W ramach dobrych praktyk procesu data science, cały scenariusz eksperymentalny powinien być w pełni odtwarzalny (reprodukowalność eksperymentów), tzn. powinno być możliwe uruchomienie eksperymentów przez prowadzących i otrzymanie takich samych wyników jak w złożonym raporcie. W tym celu można wykorzystać różne podejścia, m.in. **skrypty powłoki Bash** ("skrypty shellowe"), które będą uruchamiać skrypty Pythonowe w odpowiedniej kolejności z właściwymi argumentami. Podobne rozwiązanie można uzyskać wykorzystując tzw. **Makefile**.

Narzędziem o którym warto wspomnieć jest również **DVC (Data Version Control)**, które służy właśnie do wersjonowania kodu wraz z danymi (wykorzystując system kontroli wersji

git oraz zewnętrzne repozytoria danych - przy czym te drugie są nieobowiązkowe). Definiowane są tutaj DVC stages, które odpowiadają kolejnym etapom w przetwarzaniu danych (czyszczenie danych, wizualizacja, ekstrakcja cech, uczenie modeli, ewaluacja modeli, agregacja wyników, wizualizacja wyników itp.)

2.6 Prezentacja na zajęciach (5 pkt)

- forma dowolna (np. slajdy, notebook, dedykowana aplikacja, itd.)
- krótki i zwięzły opis projektu
- przedstawienie analizy eksploracyjnej
- przedstawienie modeli
- opis scenariusza eksperymentalnego
- przedstawienie wyników
- wnioski