Projekt zaliczeniowy

Bootcamp Data Science ING

sages

Warunkiem uzyskania certyfikatu pozytywnego ukończenia kursu jest zrealizowanie projektu końcowego. Niniejszy dokument określa zasady zaliczenia projektu.

Warunki ogólne:

1. Tematem projektu może być jeden wybrany problem z listy tematów zamieszczonej w dalszej części tego dokumentu lub inny temat określony samodzielnie (reguły dotyczące własnych tematów opisane są na końcu dokumentu).
2. Projekt realizowany jest samodzielnie.
3. Forma przekazania projektu do oceny: **projekt musi być umieszczony w repozytorium (publicznym) na githubie**. Projekt musi zawierać plik w formacie .ipynb (notebook Jupyter'owy), w którym znajdować się będzie cała realizacja projektu lub główna jego część przedstawiająca kluczowe elementy (dopuszczalne są ewentualne pliki pomocnicze pomagające zachować przejrzystość pliku głównego). Dane mogą, ale nie muszą znajdować się w repozytorium. Inne formy realizacji i przekazania projektu nie są dopuszczalne - w szczególności **przesłanie plików z rozwiązaniem nie jest traktowane jako przekazanie rozwiązania**. Link do repozytorium z projektem należy przesłać na adres: **n.ryciak@sages.com.pl .** Prosimy nie zapominać o podpisaniu się.
4. Termin oddania projektów: **14 dni od ostatniego dnia zajęć** grupy, w której uczestniczył uczestnik. Projekt oczywiście można oddać wcześniej.
5. Aby projekt został zaliczony:
    a. musi być poprawny pod względem merytorycznym (oznacza to np.: poprawną metodologicznie ocenę modeli czy właściwe przygotowanie danych, bez wyraźnych błędów w sztuce),
    b. musi spełniać wymogi określone dla danego tematu,
    c. musi mieć czytelną, przejrzystą formę umożliwiającą osobie sprawdzającej zrozumienie jego przebiegu. W szczególności **niezbędne są komentarze i objaśnienia**, które wyjaśniają poszczególne kroki. **Projekty nieestetyczne/trudne w odbiorze/bez objaśnień słownych będą zwracane do poprawki.**
6. Projekt może być zrealizowany na danych, które pojawiły się na zajęciach, ale wówczas jego zawartość musi wykraczać istotnie poza elementy zrealizowane podczas zajęć.
7. Projekt, który nie zostanie pozytywnie oceniony, zostanie zwrócony do uczestnika szkolenia wraz z komentarzem, które elementy wymagają korekty. Uczestnik szkolenia musi odesłać poprawiony projekt w ciągu 14 dni od daty przekazania przez firmę informacji o negatywnej ocenie pracy. Dalszej możliwości poprawy nie przewiduje się.

8. Projekt może zostać zaliczony „z wyróżnieniem”. Aby otrzymać wyróżnienie projekt musi być bardzo wysokiej jakości pod względem estetycznym i „organizacyjnym” (dobrze się prezentować oraz mieć wysokiej jakości kod) oraz spełniać przynajmniej jeden z warunków:
    a. Wykorzystywać zaawansowane algorytmy uczenia maszynowego wykraczające poza zakres zajęć (przykładowo: Bayesowskie metody optymalizacji parametrów, zaawansowane techniki selekcji zmiennych, zaawansowane techniki optymalizacji hiperparametrów, zaawansowane techniki redukcji wymiaru, implementacja zaawansowanych struktur sieci neuronowych).
    b. Zawierać wysokiej jakość (tzn. obiektowej, w konwencji sklearn’owej) implementację techniki/technik uczenia maszynowego. Przykładowo:
        i. Nietrywialny algorytm odnoszący się do niezbalansowanych klas.
        ii. Algorytm uzupełniania braków danych przy użyciu algorytmów uczenia maszynowego.
        iii. Algorytm selekcji zmiennych przy użyciu algorytmów genetycznych.
        iv. Klasyfikator o złożonym schemacie działania.

    Uwaga: **Objętość projektu nie przyczynia się do przyznania wyróżnienia.** Liczy się poziom zaawansowania zawartości, a nie ilość. Ponadto nie ma znaczenia czy temat jest wybrany samodzielnie czy z podanej listy.

9. Wszelkie pytania dotyczące projektu należy kierować na adres n.ryciak@sages.com.pl

Lista tematów

Temat 1

Przewidywanie ceny domów. Celem projektu jest zastosowanie modeli regresji do przewidzenia ceny domu o podanych cechach.

Link do danych: web.stanford.edu/class/stats191/data/ames2000_NAfix.csv

Dokładny opis danych: http://web.stanford.edu/class/stats191/data/amesdoc.txt

Zmienna celu (cena domu): SalePrice

Wymogi:

- Musi zostać wykorzystany model regresji liniowej.
- Muszą zostać wykorzystane przynajmniej dwie inne metody regresji: regresja Ridge, regresja Lasso, drzewo regresyjne, las losowy regresyjny, XGBoost lub inne).
- Trzeba uwzględnić optymalizację modeli/pipelinów.
- Nie usuwamy żadnych obserwacji - braki danych uzupełniamy w jakiś sposób.
- Wszelkie nieoczywiste przekształcenia danych należy opatrzyć uzasadnieniem dlaczego dokonujemy tego przekształcenia (np. robimy wykres i uzasadniamy wzięci logarytmu ze zmiennej tym, że rozkład jest skośny).
- Musi pojawić się ocena graficzna predykcji modeli (np. wykresy wartości przewidywanych od prawdziwych czy wykresy reziduów).
- Projekt musi być zakończony przejrzystym porównaniem przetestowanych rozwiązań (modeli/pipelinów) **w postaci tabeli**, gdzie jeden wiersz opisuje jedno rozwiązanie i jego wynik.

Temat 2

Przewidywanie wzięcia pożyczki. Celem projektu jest zastosowanie modeli klasyfikacji (binarnej) do przewidzenia czy pożyczka zostanie udzielona danemu klientowi z danymi parametrami wniosku.

Link do danych: https://raw.githubusercontent.com/saimadhu-polamuri/DataHakthon3X/master/dataSet/Train.csv

Opis danych (najdokładniejszy jaki istnieje): https://discuss.analyticsvidhya.com/t/hackathon-3-x-predict-customer-worth-for-happy-customer-bank/3802

Zmienna celu (wypłacenie pożyczki): Disbursed

Wymogi:

- Zmienna LoggedIn nie może być uwzględniona w modelowaniu - należy ją od razu wyrzucić ze zbioru.
- Nie usuwamy żadnych obserwacji - braki danych uzupełniamy w jakiś sposób.
- Wszelkie nieoczywiste przekształcenia danych należy opatrzyć uzasadnieniem dlaczego dokonujemy tego przekształcenia (np. robimy wykres i uzasadniamy wzięci logarytmu ze zmiennej tym, że rozkład jest skośny).
- Należy wykorzystać przynajmniej 3 metody klasyfikacji.
- Należy uwzględnić optymalizację modeli/pipelinów.
- Projekt musi być zakończony przejrzystym porównaniem przetestowanych rozwiązań (modeli/pipelinów) **w postaci tabeli**, gdzie jeden wiersz opisuje jedno rozwiązanie i jego wynik.

Temat 3

Klasyfikacja wydźwięku twittów. Celem projektu jest zastosowanie modeli klasyfikacji (binarnej) do rozpoznawania wydźwięku (pozytywny lub negatywny) twittów.

Link do danych: https://archive.ics.uci.edu/ml/machine-learning-dAAatabases/00331/sentiment%20labelled%20sentences.zip

Opis danych: https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences

Dane są podzielona na trzy pliki amazon, yelp oraz imdb - należy je połączyć i traktować jako jeden zbiór.

Dopuszczalne jest również użycie dowolnego innego zbioru tekstów z określonym wydźwiękiem. (np. twittów z serwisu kaggle).

Wymogi:

- W rozwiązaniu musi pojawić się ”czyszczenie” tekstu.
- Należy wykorzystać metody reprezentacji tekstu: macierz liczności słów, macierz TfIdf, redukcja wymiaru przy użyciu SVD (ewentualnie również inne)
- Należy wykorzystać przynajmniej 3 metody klasyfikacji.
- Należy uwzględnić optymalizację modeli/pipelinów.
- Projekt musi być zakończony przejrzystym porównaniem przetestowanych rozwiązań (modeli/pipelinów) **w postaci tabeli**, gdzie jeden wiersz opisuje jedno rozwiązanie i jego wynik.

Temat 4

Klasyfikacja obrazów. Celem projektu jest zastosowanie algorytmów do klasyfikacji do problemu rozpoznawania rasy psów przedstawionych na zdjęciu.

Link do danych: vision.stanford.edu/aditya86/ImageNetDogs/images.tar

Opis danych: http://vision.stanford.edu/aditya86/ImageNetDogs/

Dane zawierają zdjęcia psów 120 różnych ras. Jeżeli ilość danych będzie powodować problemy z czasem obliczeń można sobie ten zbiór zmniejszyć - np. wziąć zdjęcia tylko 10-ciu ras i na takim zbiorze pracować.

Wymogi:

- Jeżeli projekt będzie wykorzystywał algorytmy klasycznego uczenia maszynowego, to

    - należy porównać działanie modeli na danych surowych oraz danych w jakiś sposób przekształconych/cechach wygenerowanych przez nas
    - należy wykorzystać przynajmniej 3 metody klasyfikacji.
    - należy wówczas również uwzględnić optymalizację modeli/pipelinów.
- Jeżeli projekt będzie wykorzystywał konwolucyjne sieci neuronowe, to trzeba rozpatrzeć kilka różnych struktur/wariantów sieci.
- Projekt musi być zakończony przejrzystym porównaniem przetestowanych rozwiązań (modeli/pipelinów) **w postaci tabeli**, gdzie jeden wiersz opisuje jedno rozwiązanie i jego wynik.

Własny temat

Wymogi:

- Projekt musi być nietrywialny - w projekcie musi być widoczny istotny wkład pracy uczestnika. Musi dowodzić umiejętności pracy z danymi i znajomości uczenia maszynowego. Musi mieć wartość merytoryczną - pokazywać rozwiązanie jakiegoś problemu lub dostarczać wyników analiz. W bardzo dużym uproszczeniu: nie może polegać na tym, że wczytane są czyste dane oraz jest dopasowany i oceniony model.
- Projekt musi skupiać się na uczeniu maszynowym (czyli celem projektu nie może być np. napisanie aplikacji do wizualizacji danych).
- Dane do projektu uczestnik wybiera samodzielnie (np. z serwisu kaggle.com)
- Projekt musi mieć przejrzystą strukturę, być uporządkowany i estetyczny. W szczególności musi zawierać:
    - opis celu projektu
    - opis danych
    - klarowne objaśnienia przetwarzania danych - osoba czytająca projekt musi z komentarzy dowiedzieć się jak wyglądają dane po przekształceniach.

- Projekt musi zawierać elementy „porównawcze” (przynajmniej jeden z wymienionych punktów lub inne porównania):
    - porównanie różnych modeli
    - porównanie różnych reprezentacji danych
    - porównanie różnych sposobów przetworzenia danych
- Projekt może skupić się np. na rozwiązaniu problemu klasyfikacji czy regresji, ale celem może być również nie samo rozwiązanie problemu, ale przeprowadzenie analiz, takich jak:
    - Porównanie różnych technik dla niezbalansowanych danych.
    - Porównanie różnych metod optymalizacji hiperparametrów (metod wykraczających poza zakres zajęć).
    - Analiza porównawcza różnych metod oceny istotności zmiennych.
    - Porównanie różnych metod reprezentacji danych tekstowych w jakimś zadaniu.
- Projekt musi być zakończony przejrzystym porównaniem przetestowanych rozwiązań/modeli/pipelinów **w postaci tabeli**, gdzie jeden wiersz opisuje jedno rozwiązanie i jego wynik.