project

April 7, 2025

0.0.1 Ogólne podpowiedzi do wszelkich dalszych raportów:

1. Najważniejsze w każdej odpowiedzi są interpretacje uzyskanych rezultatów, wnioski i uzasadnienia. Zamieszczenie rezultatów liczbowych służy uzasadnieniu wniosków; surowe rezultaty bez interpretacji autora są bezwartościowe. Zatem nie stosujemy takiego podejścia: "Zamieściłem wykres i widzę na nim, że trafność rośnie, więc nie muszę tego pisać" albo "Podałem dwie liczby i widzę z nich, że jedna klasa jest 5x bardziej liczna od drugiej, więc nie muszę tego pisać". Nie podajemy też samych wniosków tekstowych bez podparcia konkretnymi wynikami.
2. Wykresy zwykle lepiej i zwięźlej pokazują wyniki, niż duże tabele liczb.
3. Aby uniknąć pustych marginesów na wykresach, do każdego z nich używaj tight layout.
4. Nadawaj informatywne nazwy plikom z obrazkami ("drzewa_dec_trafnosc.png" zamiast "wykres3.png" albo "Download_18.png").
5. Podając liczby zwracaj uwagę na odpowiednia (uzasadnioną) liczbę miejsc znaczących – zwykle nie potrzeba 5, 10, a tym bardziej 15 miejsc po przecinku.
6. Unikaj nieuzasadnionych, subiektywnych określeń ("dużo", "bardzo słabe") – żeby podeprzeć takie oceny, podawaj również konkretne wartości.
7. Wyrażaj się precyzyjnie i jednoznacznie; używaj terminologii uczenia maszynowego ("atrybuty" zamiast "kolumny", "przypadek" zamiast "element").
8. Unikaj mieszania języków ("clustrowanie", "model overfituje", "w drzewie były dwa splity", "przypadki nie mają labeli", "dane olabelowane", "zaawansowane setupy", "wartości zostały przeprocesowane", "w tym datasecie") – jeśli koniecznie chcesz użyć angielskiego terminu, bo nie ma dobrego polskiego odpowiednika, nie odmieniaj go i pisz takie wyjątkowe słowa italikiem ("zachodzi overfitting" – chociaż tu akurat jest dobry odpowiednik).
9. Przygotowuj odpowiedzi samodzielnie (trudno "odzobaczyć" to, co już się zobaczyło – ryzyko plagiatu).
10. Kopiowanie i wklejanie na eKursach: jeśli nie działa Ctrl-C/Ctrl-V, spróbuj Ctrl-Insert/Shift-Insert.

0.0.2 Zadanie 1.

Przypomnij sobie z wykładów oraz z wcześniejszych przedmiotów nawiązujących do tematyki uczenia maszynowego i analizy danych, jakie techniki służą do rozwiązania zadania regresji dla wielowymiarowych danych. To zadanie nie podlega ocenie; zastanów się i wpisz tutaj nazwy wszystkich algorytmów, które przychodzą Ci do głowy.

0.0.3 Odpowiedź 1.

- Regresja liniowa

- k-NN
- Drzewa decyzyjne
- Random Forest
- SVR
- Sieci neuronowe

0.0.4 Zadanie 2.

Pobierz zbiór danych o nazwie odpowiadającej Twojemu numerowi albumu. Te dane dotyczą wykrywania anomalii (zakłóceń) w sygnale audio; każdy wiersz opisuje inne wystąpienie anomalii, a ostatnia kolumna to szerokość zakłócenia (liczba próbek). Pozostałe kolumny to różne statystyki zebrane z otoczenia zakłócenia; pierwszy wiersz zawiera skrótowe nazwy kolumn. Szczegółowy opis znaczenia atrybutów znajdziesz tutaj. Możesz wczytać plik używając dane = np.genfromtxt(nazwa_pliku, skip_header=1) albo parametru names=True (wtedy uwaga).

Przeprowadź jego wstępną eksplorację: liczba i rodzaje atrybutów, ich zakresy i rozkłady wartości. Pokaż rozkłady wartości wszystkich atrybutów warunkowych obok siebie na jednym szerokim wykresie pudełkowym lub skrzypcowym; na osi poziomej umieść nazwy atrybutów. Opisując wnioski (wystarczy kilka zdań) możesz pogrupować (o ile to możliwe) atrybuty pisząc np. “73 atrybuty są takie a takie, 22 atrybuty charakteryzują się tym a tym, wyjątkowy jest atrybut taki a taki”, itp.

0.0.5 Odpowiedź 2.

0.0.6 Zadanie 3.

Przejrzyj dostępne metryki oceny modeli regresji. Które z nich wydają Ci się łatwe do interpretacji i dlaczego? Weź pod uwagę konkretny problem, którym się zajmujemy (predykcja ostatniej kolumny w zbiorze i znaczenie tej kolumny). Wybierz dwie metryki, które Twoim zdaniem niosą użyteczną informację o jakości modelu w rozpatrywanym problemie (jeśli masz ochotę, możesz wybrać więcej niż dwie). W kolejnych pytaniach oznaczam te metryki jako M1 i M2. Uzasadnij swój wybór.

0.0.7 Odpowiedź 3.

0.0.8 Zadanie 4.

Do dalszych testów użyjemy następujących technik:

```
from sklearn import linear_model
from sklearn import neighbors  # KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.svm import SVR
```

Przejrzyj dokumentację scikit-learn i napisz, jakie jeszcze dostępne tam i znane Ci metody mogłyby posłużyć do zbudowania modeli regresji.

0.0.9 Odpowiedź 4.

0.0.10 Zadanie 5.

Porównaj metody wymienione w treści poprzedniego pytania (jeśli masz ochotę, możesz dodatkowo przetestować jeszcze inne) pod kątem M1 oraz M2 dla całego zbioru danych (bez podziału uczący–testujący). Użyj domyślnych wartości parametrów (jeśli masz ochotę, poeksperymentuj z doborem parametrów; użyj też nie-domyślnych wartości wtedy, kiedy uważasz, że domyślne wartości nie mają sensu w tym zastosowaniu lub są niepoprawne). Dla SVR porównaj kernel liniowy i RBF. Dla drzew decyzyjnych użyj max_depth=2 (co się dzieje, kiedy nie ograniczymy głębokości?).

Załącz dwa wykresy (jeden dla M1 i jeden dla M2) porównujące powyższe metody. Opisz wnioski.

Fragmenty kodu, które mogą się przydać:

```
print(dane.shape)
Xregr=dane[:,0:-1]
yregr=dane[:,-1]
from sklearn.metrics import mean_absolute_error  # przykład dla R2 i MAE
print_metrics = lambda regresor,opis,X,y: print(opis,':  R^2=%.2f,'%regresor.score(X, y),
                                    ' MAE=%.1f'%mean_absolute_error(y, regresor.predict(X)))
regr = neighbors.KNeighborsRegressor()
regr.fit(Xregr, yregr)
print_metrics(regr,'KNeighborsRegressor',Xregr,yregr)
```

0.0.11 Odpowiedź 5.

0.0.12 Zadanie 6.

Które z metod wykorzystanych w poprzednim zadaniu wymagają normalizacji/standaryzacji danych i nie powinniśmy ich używać na surowych danych? Dlaczego tak jest, w czym tkwi niebezpieczeństwo? Rozszerz wykresy z poprzedniego zadania o wyniki poprawnie użytych metod oraz zinterpretuj efekt wykorzystania normalizacji.

Fragmenty kodu, które mogą się przydać:

```
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler  # albo inny, bardziej odpowiedni
regr = make_pipeline(StandardScaler(), SVR(ewentualniejakiesparametry))
```

0.0.13 Odpowiedź 6.

0.0.14 Zadanie 7.

Wybierz Twoim zdaniem najlepszy model regresji. Napisz, na jakiej podstawie go wybrałeś/wybrałaś i co nam daje takie kryterium "najlepszości". Załącz wykres, w którym na osi poziomej są faktyczne wartości zmiennej zależnej, a na osi pionowej – to, co przewiduje wybrany model regresji. Postaraj się, żeby wykres był czytelny i przydatny (zamiast zaproponowanego wykresu możesz przygotować inny, który lepiej pokaże błędy popełniane przez model na poszczególnych przypadkach) oraz opisz wnioski z tej wizualizacji.

0.0.15 Odpowiedź 7.

0.0.16 Zadanie 8.

Który z wytworzonych modeli regresji jest najlepiej interpretowalny dla człowieka? Spróbuj go zwizualizować (sam model) i zinterpretuj, jak on działa (jego "wiedzę"); możesz tutaj wykorzystać specjalnie dobrane wartości parametrów, żeby wytworzyć jeszcze lepiej interpretowalny model bez dużej utraty jego jakości.

Zostaw sobie na przyszłość komentarze w kodzie; ten model i dane będą jeszcze używane na ostatnich zajęciach.

0.0.17 Odpowiedź 8.

0.0.18 Zadanie 9.

Oceń zdolność predykcji modeli tego samego rodzaju (te same algorytmy i wartości parametrów), co utworzone wcześniej, używając 10-krotnej kroswalidacji (uwaga). Sporządź i załącz dwa analogiczne wykresy (M1 i M2; możesz pokazać obok siebie wartości tych metryk dla całego zbioru i średnie z kroswalidacji). Czy te rodzaje modeli, które najlepiej sprawdzały się dla całego zbioru danych to te same rodzaje, które najlepiej przewidują wartości atrybutu decyzyjnego na zbiorze testowym?

Przed wysłaniem całego quizu przejrzyj jeszcze raz listę podpowiedzi z pytania 1.

0.0.19 Odpowiedź 9.