Statystyczna Analiza Danych
Projekt zaliczeniowy 2024

1 Informacje wstępne

1.1 Opis danych

Narządy, takie jak trzustka, składają się z wielu typów tkanek, a te z kolei z wielu typów komórek. W obrębie trzustki możemy wyróżnić komórki typowe wyłącznie dla tego narządu, takie jak komórki alfa czy beta, ale także komórki związane z ukrwieniem czy układem immunologicznym.
Dane w tym zadaniu pochodzą z wielomodalnego sekwencjonowania pojedynczej komórki (ang. *multimodal single cell RNA sequencing,* ***scRNA-seq***). Użycie *scRNA-seq* pozwala na studiowanie próbek w wysokiej rozdzielczości i oddzielenie od siebie komórek różnych typów. Możliwe jest między innymi porównanie komórek patologicznych, pobranych od pacjentów nowotworowych, z komórkami zdrowymi. W technologii multimodal *scRNA-seq* dla każdej komórki otrzymujemy dwa typy odczytów:

- **Zliczenia transkryptów RNA** odpowiadające ekspresji (aktywności) genów w danej komórce;
- **Ilość białek powierzchniowych (ang. *protein abundance*)**, która jest wprost związana z typem danej komórki.

Wynikiem eksperymentu *scRNA-seq* są macierze, w których dla każdej komórki przypisany jest sygnał RNA z wielu tysięcy genów (w naszym zadaniu X) oraz sygnał pochodzący z kilkudziesięciu białek powierzchniowych (w naszym zadaniu dla uproszczenia wybraliśmy pojedyncze białko CD36, y).
Zgodnie z centralnym dogmatem biologii, wiemy, że informacja genetyczna przepływa z RNA na białka. Tym samym, należy spodziewać się korelacji między ilością białka a ekspresją genu, który to białko koduje. Z przyczyn technicznych i biologicznych, ta zależność niejednokrotnie ulega degeneracji. Problem w tym zadaniu polega na predykcji sygnału z białek powierzchniowych na podstawie ekspresji genów. Przewidywanie sygnału *protein abundance* jest kluczowe dla większości publicznie dostępnych zbiorów, dla których dostępna jest wyłącznie macierz RNA. Analiza sygnału o ekspresji genów i ilości białek powierzchniowych znacząco ułatwia proces identyfikowania i nazywania komórek w próbce.
Dane zostały pobrane z szpiku kostnego ludzkich dawców. Zebrane komórki to w większości komórki układu immunologicznego. Prawidłowe zidentyfikowanie limfocytów typu T w oparciu o oba typy odczytów w zbiorze takiego typu mogłoby być podstawą do rozwijania celowanych terapii nowotworowych (dla ciekawych: *CAR T cell therapy*).

1.2 Sposób pobrania danych

Na przedmiotowej stronie Moodle znajduje się link do folderu z danymi dla każdej z grup laboratoryjnych. Dane można również pobrać z Kaggle.

Ponieważ każda grupa pracuje na danych pochodzących z innego eksperymentu, wyniki pomiędzy grupami mogą się różnić. Dane są skompresowane oraz zapisane w formacie `.csv`. Udostępnione będą trzy pliki (uwaga, mogą być skompresowane przy użyciu programu **gzip**)

- `X_train.csv` oraz `X_test.csv`, zawierające macierze RNA. Każdy wiersz odpowiada komórce, kolumna genowi, natomiast wartości to poziom ekspresji. Kolumny tych macierzy to nasze zmienne objaśniające.
- `y_train.csv`, odpowiadający ilości białka powierzchniowego pewnego typu w komórkach (tych, których dotyczyły dane z pliku `X_train.csv`). Jest to nasza *zmienna objaśniana*. W dalszej części opisu, dane z plików `X_train.csv` i `y_train.csv` będziemy nazywać treningowymi, a dane z pliku `X_test.csv` będziemy nazywać testowymi.

1.3 Sposób oddania projektu

Pliki wyspecyfikowane niżej należy wysłać mailem na adres prowadzącego swojej grupy laboratoryjnej. Predykcję (rozwiązanie zadania 6., czytaj dalej) należy także zgłosić do Kaggle. **Link do konkursu Kaggle zostanie udostępniony na stronie Moodle przedmiotu**. W ramach Kaggle każdy student będzie mógł zgłaszać wiele propozycji predykcji i w ten sposób uzyskiwać pewne informacje o jej jakości, a także jej aktualną pozycję w rankingu.
Pliki do wysłania prowadzącemu:

- Dla zadań 1-5: raport w formacie `.pdf` lub `.html`, realizujący opisane w treści zadań polecenia (szablon nazwy pliku: `NrIndeksu_raport.ext`, gdzie `ext` to odpowiednie rozszerzenie).

  Uwaga! Proszę nie stosować znaczników ukrywających kod, który generuje kolejne fragmenty raportu. Kod jest ważniejszy niż ostateczny efekt wizualny.

- Dla zadań 1-5: kod zródłowy (np. `.Rmd` lub `.R`) generujący rozwiązania zadań (szablon nazwy pliku: `NrIndeksu_kod.ext`, gdzie `ext` to odpowiednie rozszerzenie).

- Dla zadania 6.: wyniki predykcji na danych testowych w formie pliku `.csv`, zawierającego kolumnę *Id* z numerami obserwacji oraz kolumnę *Expected* z wartościami predykcji (szablon nazwy pliku: `NrIndeksu_predykcja.csv`).

  Przykładowy plik z predykcją zawierający losowe dane został udostępniony na przedmiotowej stronie Moodle.

  Obowiązkowy dla wszystkich studentów jest udział w Kaggle. Plik z predykcją przesłany do oceny prowadzącego musi również zostać zgłoszony (przed upłynięciem terminu) do Kaggle. **Zwracamy uwagę, że do Kaggle można wysłać więcej niż jedną predykcję nawet znacznie przed terminem, do czego zachęcamy**. Prowadzący ma prawo przyznać punkty w zadaniu 6. jedynie na podstawie tzw. *leaderboard* w Kaggle, lub jedynie na podstawie przesłanych do niego predykcji, zależnie od swoich preferencji. Zachęcamy studentów, by w każdej grupie wyjaśnili to z prowadzącym.

  W zadaniu 6. należy dosłać również raport PDF i kod źródłowy, nie mamy jednak tak kategorycznych wytycznych co do ich postaci. W szczególności, kod źródłowy nie musi być w `R`. Satysfakcjonujący poziom kodu i raportu to warunek konieczny przyznania punktów w zadaniu 6., ale nie jest to warunek dostateczny. Punkty otrzymuje się za niski błąd predykcji i wyjaśniono to szczegółowo w opisie zadania.

1.4 Ocena

Za cały projekt można otrzymać 30 punktów. Maksymalne liczby punktów za każde zadanie podane są w nawiasach wraz z treścią zadań w rozdziale 2. Ocena zadań będzie uwzględniała

- realizację przedstawionych poleceń,
- jakość raportu w formie .pdf lub .html (wizualizacje, czytelność tekstu, opis wyników),
- jakość wykorzystanego w tym celu kodu. Warto zadbać o to, by był on czytelny i reprodukowalny. Warto zadbać też o to, by **nie** był ukryty przy użyciu specjalnych znaczników.

Dodatkowe informacje na temat szczegółów punktacji można uzyskać u swojego prowadzącego laboratorium.
W przypadku zadania 6., prowadzący ma prawo przyznać punkty jedynie na podstawie tzw. *leaderboard* konkursu Kaggle, lub jedynie na podstawie przesłanych do niego plików z predykcją, zależnie od swoich preferencji. Prosimy o wyjaśnienie tego z prowadzącym swojej grupy laboratoryjnej. Niezależnie od decyzji prowadzącego, udział w Kaggle jest dla wszystkich studentów obowiązkowy. Przesłanie prowadzącemu raportu oraz kodu źródłowego dla tego zadania są warunkiem koniecznym, ale nie dostatecznym, by dany student był oceniany w tym zadaniu. Punkty w zadaniu 6. otrzymuje się za niski wynik błędu predykcji mierzony RMSE. Szczegóły przyznawania punktów zostały opisane w treści zadania.

1.5 Terminy

- Terminem na oddanie rozwiązań zadań 1-5 jest 12 maja 2024 (niedziela), godzina 23:59
- Terminem na oddanie rozwiązania zadania 6 jest 2 czerwca 2024 (niedziela), godzina 23:59

1.6 Zadania oddane po terminie

- Rozwiązania zadań nadesłane po terminie będą oceniane o 10% niżej za każdą **rozpoczętą** godzinę opóźnienia.
- Rozwiązanie nadesłane 10 godzin lub więcej po terminie nie będą brane pod uwagę przy ocenie. Otrzymają 0 punktów.

Uwaga! Aby uniknąć przkroczenia terminu dla zadania 6, warto na bieżąco wrzucać nowe pliki z coraz to lepszymi predykcjami na Kaggle.
Uwaga! Konkurs Kaggle zostanie zamknięty wraz z upłynięciem terminu oddawania rozwiązań zadania 6.; po tym terminie warto rozważyć, czy zysk z lepszego rozwiązania przewyższa stratę punktów za rozwiązanie po terminie. Jeśli tak, rozwiązanie po terminie proszę wysyłać tylko do prowadzącego, z pominięciem Kaggle.

2 Treści zadań

1. **Eksploracja** (*3 pkt.*)

   (a) Sprawdź, ile obserwacji i zmiennych zawierają wczytane dane treningowe oraz testowe. Przyjrzyj się typom zmiennych i, jeśli uznasz to za słuszne, dokonaj odpowiedniej konwersji przed dalszą analizą. Upewnij się, czy dane są kompletne.

   (b) Zbadaj rozkład empiryczny zmiennej objaśnianej (przedstaw kilka podstawowych statystyk, do analizy dołącz histogram lub wykres estymatora gęstości).

   (c) Wybierz 250 zmiennych objaśniających najbardziej skorelowanych ze zmienną objaśnianą. Policz korelację dla każdej z par tych zmiennych. Zilustruj wynik za pomocą mapy ciepła (ang. *heatmap*).

   Uwaga! Opisany tu wybór zmiennych jest tylko na potrzeby niniejszego podpunktu, analizę opisaną w kolejnych zadaniach należy przeprowadzić na **pełnym** zbiorze danych treningowych.

2. **Testy statystyczne** (*6 pkt.*)

   (a) Narysuj wykres kwantylowy porównujący zmienną objaśnianą z rozkładem normalnym. Na wykresie zaznacz prostą wyznaczoną przez kwantyle doświadczalne. Odpowiedz na pytanie, czy da się odczytać średnią i wariancję rozkładu doświadczalnego wprost z tego wykresu?

   (b) Przeprowadź test statystyczny hipotezy zgodności zmiennej objaśnianej z rozkładem normalnym.

   (c) Wybierz zmienną objaśniającą najbardziej skorelowanej ze zmienną objaśnianą

      i. Przeprowadź test statystyczny hipotezy zgodności wybranej zmiennej objaśniającej z wybranym (sensownym!) rozkładem (np. z rozkładem wykładniczym z parametrem 10, jeśli są ku temu przesłanki).

      ii. Wybierz test statystyczny i sprawdź, czy wybrana zmienna objaśniająca ma podobny rozkład w zbiorze testowym i treningowym (w tym podpunkcie nie chodzi o test zgodnosci, ale o test sprawdzający lokalizację tych dwóch rozkładów - wzajemne położenie średnich lub median).

   Uwaga! Przy rozwiązaniu tego zadania punktowane jest sformułowanie hipotezy zerowej, hipotezy alternatywnej, podanie poziomu istotności, wybranie (i uzasadnienie wyboru) testu, przeprowadzenie samego testu i omówienie jego wyniku.

3. **ElasticNet** (*6 pkt.*)

   Pierwszy model, który należy wytrenować, to *ElasticNet*. Podczas wykładu spotkaliśmy się z jego szczególnymi przypadkami: regresją grzbietową (ang. *ridge regression*) oraz lasso.

   (a) Wyszukaj i przedstaw w raporcie informacje o modelu ElasticNet. Opisz parametry, które są w nim estymowane, optymalizowaną funkcję oraz hiperparametry, od których ona zależy. Dla jakich wartości hiperparametrów otrzymujemy regresję grzbietową, a dla jakich lasso?

(b) Zdefiniuj siatkę (ang. *grid*) hiperparametrów, opartą na co najmniej trzech wartościach każdego z hiperparametrów. Zadbaj o to, by w siatce znalazły się konfiguracje hiperparametrów odpowiadające regresji grzbietowej i lasso. Użyj walidacji krzyżowej do wybrania odpowiednich hiperparametrów (o liczbie podzbiorów użytych w walidacji krzyżowej należy zdecydować samodzielnie oraz uzasadnić swój wybór).

(c) Narysuj wykres skrzypcowy (ang. *violin plot*) dla błędów średniokwadratowych, otrzymanych w poszczególnych foldach testowych walidacji krzyżowej (wartości na osi Y) dla danego zestawu wartości hiperparametrów (na osi X). Uzyskane wartości błędów średniokwadratowych w foldach powinny być również oznaczone na wykresie jako punkty. Dobrze przemyśl sposób prezentacji: użyj kolorów, umieść legendę lub wyczerpujący opis.

(d) Podaj błąd treningowy i walidacyjny modelu dla wybranych wartości hiperparametrów (należy uśrednić wynik względem wszystkich podzbiorów testowych wyróżnionych w walidacji krzyżowej).

4. **Lasy losowe** (*6 pkt.*) W tej części projektu należy wytrenować model lasów losowych.

   Uwaga! Należy użyć tych samych foldów walidacji krzyżowej co poprzednio.

   (a) Spośród wielu hiperparametrów charakteryzujących model lasów losowych wybierz trzy różne. Zdefiniuj trójwymiarową siatkę przeszukiwanych kombinacji hiperparametrów i za pomocą walidacji krzyżowej wybierz ich optymalne (w kontekście wykonywanej predykcji) wartości. Wykorzystany przy walidacji krzyżowej podział danych powinien być taki sam, jak w przypadku ElasticNet.

   (b) Narysuj wykres pudełkowy (ang. *box plot*) dla błędów średniokwadratowych, otrzymanych w poszczególnych foldach testowych walidacji krzyżowej (wartości na osi Y) dla danego zestawu wartości hiperparametrów (na osi X). Dobrze przemyśl sposób prezentacji: użyj kolorów, umieść legendę lub wyczerpujący opis.

   (c) Podaj błąd treningowy i walidacyjny modelu dla wybranych wartości hiperparametrów (należy uśrednić wynik względem wszystkich podzbiorów testowych wyróżnionych w walidacji krzyżowej).

5. **Podsumowanie** (*2 pkt.*)

   Zrób podsumowanie tabelaryczne wyników, jakie otrzymywały metody w walidacji krzyżowej w obu rozważanych powyżej modelach (tzn. ElasticNet oraz lasy losowe). Porównanie to jest powodem, dla którego zależy nam na zastosowaniu tych samych podziałów walidacji krzyżowej.

   Do porównania dołącz trzeci model, referencyjny, również poddany walidacji krzyżowej, tzn. model, który dowolnym wartościom zmiennych objaśniających w foldzie testowym przypisuje średnią arytmetyczną zmiennej objaśnianej *policzoną w foldach treningowych*.

   Określ, który model wydaje Ci się najlepszy (uzasadnij swój wybór).

6. **Predykcja na zbiorze testowym** (*7 pkt.*)

   Ta część projektu ma charakter otwarty oraz **późniejszy termin oddawania**. Można ją zaimplementować w dowolnym języku programowania, np. w `R` lub w `Python` lub w

`C++`. W oparciu o dane treningowe należy dopasować dowolnie wybrany model, a następnie zastosować go do przewidywania wartości zmiennej objaśnianej w zbiorze testowym. Sposób wyboru i budowy modelu, a także motywacje stojące za takim wyborem powinny zostać opisane w raporcie PDF lub HTML, dotyczącym tej części. Do spisania raportu w tej części można użyć LaTeX (nie musi to byc raport `.Rmd`). Wygenerowane predykcje należy wysłać **wraz z kodem źródłowym i raportem** do prowadzącego. Predykcję należy także umieścić na stronie Kaggle (odpowiedni link zostanie udostępniony na Moodle). Liczba uzyskanych punktów zależeć będzie od jakości predykcji, mierzonej pierwiastkiem błędu średniokwadratowego, RMSE. Punkty zostaną przyznane pod warunkiem, że kod źródłowy i raport są satysfakcjonującej jakości.

Wskazówka: Warto rozważyć modele zaimplementowane w pakiecie `R`: **`caret`**. Polecamy również wypróbować różne techniki wspomagające uczenie maszynowe (np. odpowiedni dobór podzbioru zmiennych objaśniających, transformacje zmiennych lub redukcja wymiaru).

Szczegóły punktacji:

(1 pkt.) – za błąd niższy od pochodzącego z podstawowego modelu referencyjnego, który dowolnym wartościom zmiennych objaśniających przypisuje średnią arytmetyczną zmiennej objaśnianej po cały zbiorze treningowym (**`mean_baseline`** na Kaggle).

(2 pkt.) – za błąd niższy od pochodzącego z modelu ElasticNet wytrenowanego przez prowadzących laboratoria (**`enet_baseline`** na Kaggle).

(4 pkt.) – punktacja ta obliczana jest według wzoru $\frac{1}{4}\left\lfloor 16 \cdot \hat{F}(e)\right\rfloor$, gdzie e to błąd testowy predykcji studenta, $\hat{F}$ jest dystrybuantą empiryczną *ujemnych* błędów (im wyższy błąd, tym *niższa* wartość dystrybuanty empirycznej) wszystkich zgłoszonych przez studentów predykcji (w ograniczeniu do grupy laboratoryjnej studenta), natomiast $\lfloor\cdot\rfloor$ to część całkowita.

Przykładowo: osoba, która uzyska najniższy błąd ze wszystkich w swojej grupie laboratoryjnej, ma wartość dystrybuanty empirycznej 1.0, a zatem otrzyma 4 punkty zgodnie ze wzorem powyżej, a jeśli tylko jej wynik jest również niższy niż oba baseline'y, otrzyma łącznie 7 punktów.

Do oszacowania liczby uzyskanych punktów za to zadanie (z uwzględnieniem wyników innych osób w grupie) można użyć przykładowego arkusza kalkulacyjnego, który zostanie opublikowany na stronie Moodle przedmiotu.