Raport walidacyjny

Autorzy - Grupa 1

Paweł Pozorski, Michał Pytel

Poniższy dokument zawiera wyniki walidacji pracy zespołu numer 5. Zespół badawczy podjął się przewidywania kolumny Outcome z dostarczonego zbioru danych, będącą flagą z 3 możliwymi wartościami indykującymi stan zdrowia reprezentowanej osoby w kontekście wystąpienia diabieties. Proces walidacji był podzielony na 3 kroki i przeprowadzony na niedostępnym dla badaczy zbiorze danych, po każdym z 3 wymienionych poniżej kroków zespół oceniany wnosił zgłoszone przez nas poprawki.

Warto wspomnieć o ogólnych uwagach do pracy zespołu, zanim przejdziemy do „szczegółów” – do poprawy na pewno pozostaje struktura logiczna projektu. Z obecnej postaci trudno wywnioskować proces decyzyjny i na podstawie czego kluczowe kroki zostały podjęte, ponieważ kod jest nieskomentowany oraz chaotyczny. Nie ma to jednak wpływu na finalną wydajność modelu.

Etap EDA

Na tym etapie zespół badawczy stara się zgłębić i zrozumieć strukturę wewnętrzną danych. Analiza została przeprowadzona wystarczająco głęboko, wszystkie najważniejsze kroki zostały wykonane. Wykresy były informatywne, po rozmowie z zespołem my [zespół walidacyjny] uważamy że ich wnioski są porawne i replikowalne.

Korekty wymagają następujące elementy – odnośnie Project1.ipynb

1. Problem opisany po angielsku, a wnioski po polsku
2. Macierz korelacji - jest symetryczna i przekątna nie niesie informacji, dla czytelności

fajnie to zakryć i zostawić tylko dolny / górny trójkąt

1. W waszym badaniu wychodzi ze wiele zmiennych binarnych przyjmuje dominująco wartość true / false, może warto przemyśleć podział zbioru na test, train, val gwarantujący jak najlepsze rozłożenie tych wartości na każdy ze zbiorów, tj żeby nie doszło do sytuacji gdzie feature x mającą 95% wartości na true będzie w zbiorze

treningowym przyjmowała tylko wartości true etc

1. Histogramy mogły by być wyraźniejsze, m.in bez pionowego grida, no i podpis osi na pewno w tych binarnych mógłby być lepszy – programista zrozumie, ale do biznesowej prezentacji to już nie są dobre wykresy

Etap FEATURE ENGINEERING

W tym etapie zespół 5 poszukuje możliwie przydatnych nowych kolumn. W finalnej wersji projektu znajduje się próba stworzenia kilku, z czego niewielka część okazuje się przydatna, gdzie przydatność definiujemy poprzez silną korelację ze zmienną objaśnianą. Cały proces eksploatacyjny zważywszy na dostępne dane jest w naszej opinii wystarczająco głęboki, jedyną uwagą z naszej strony jest brak sprawdzenia wpływu tych zmiennych na finalną predykcję. Mamy tu na myśli nawet prosty check – trenowanie modeli z tymi zmiennymi i bez nich oraz porównanie wyników, czy stworzone kolumny zamiast wspierać predykcję jej nie zaburzają. Występuje tu później (etam MODELING) analiza sharp, jednak brak jakichkolwiek wniosków do niej.

Korekty wymagają następujące elementy – odnośnie Project1.ipynb

1. Warto pomyśleć nad pipelineami, zamiast funkcji create new features
2. Accuracy to nie jest dobra metryka oceny dokładności jeżeli jest niewyrównany target value
3. Wasze modele w większości przewidują wartość 0 przez co accuracy jest wysokie, ponieważ jest w większości dobrze trafia w większość, kiedy w rzeczywistości F1 score jest niski, spróbujcie również inne metryk
4. Potrzebne jest wyjaśnienie, na jakiej podstawie kolumny zostały stworzone w taki sposób, czy jest to ekspercka wiedza, czy za pomocą jakiegoś modelu

Etap MODELING

Słowem wstępu osiągnięte przez zespół badawczy wyniki na danych walidacyjnych są replikowane na dane testowe. Ogólne uwagi zespołu walidującego do tego kroku jest słabe przeszukanie przestrzeni parametrów dla niektórych modelów oraz brak podjętej próby sprawdzania modeli uczenia głębokiego, co jednak nie jest samo w sobie błędem.

Korekty wymagają następujące elementy – odnośnie Best\_Models.ipynb, Model\_Selection.ipynb i Models\_Tuning.ipynb

1. feature\_engineering\_pipeline – wszystkie operacje sekwencyjne, można by część z nich robić równolegle. Część z transformacji polega na hard-codowanych wartościach, nie wiadomo skąd zostały wzięte. Do deployu produkcyjnego brakuje zapisania pipeline z możliwością wczytania go z innego miejsca, aktualna struktura (funkcyjna) transformatorów nie pozwoli na łatwy zapis.
2. Dla przejrzystości, odnosi się to tak naprawdę do wszystkich notebookow, logikę pythona odpowiadającą za rysowanie wykresów itp. Wypadało by przerzucić do plików .py i jedynie importować. Notebook byłby prostszy w odszyfrowywaniu jakby był bardziej biznesowy
3. Brakuje jakiś wniosków – np. dla Multinomial Logistic Regression widzimy ze tuning nic nie daje oraz ze model gorzej radzi sobie z klasą 1. To samo można powiedzieć o pozostałych modelach. Dodatkowo przestrzeń parametrów dla większości z nich jest przeszukiwania pobieżnie, jedynie dla paru hard-codowanych wartości. Fajnie to zoptymalizować i szukać w przestrzeni parametrów ciągłych według jakieś funkcji kosztu pozwalającej określić minimum (tj najlepszą wartość parametrów), może być to droższe obliczeniowo chodź niekoniecznie – nieobiecujące przypadki można wywalać zawczasu.
4. Podkreślenie punktu 3 – cały ModelTuning.ipynb jest strasznie powtarzalny i mało czytelny – brakuje zestawienia modeli ze sobą, wniosków z ich wyników, propozycji który model ostatecznie wybrać czy tłumaczenia dlaczego optymalizujemy akurat ten – optymalizacja parametrów jest decyzją z punktu widzenia biznesu często drogą, po co „forcem” przeszukiwać parametry wszystkich dostępnych sklearnowych modeli jak w przypadku niektórych z góry widać że nie będą one „optymalne”?
5. 3 i 4 tak samo dla Model Selection.ipynb + czemu jest podział na 2 pliki o podobnych nazwach? Domyślam się ze robiły to inne osoby ale prowadzi to do niespójności
6. Jest parę fajnych analiz np. SHARP, ale cały step 3 projektu nie prowadzi do jego biznesowego zamknięcia – odnosi się wrażenie ze z całości idziemy w wiele stron, żadnej strony nie komentujemy i nie faworyzujemy – poszukiwania bez celu
7. Nazewnictwo plików nie spójne. Jako data-scientist walidujący pracę zespołu trzeba się domyślać co było po czym + brak finalnego produktu i prezentacji. Projekt robi wrażenie work-in-progress czy tez playgroundu.
8. Braki w pliku requirements.txt, trzeba wiele paczek doinstalować ręcznie

Dodatkowo przejdźmy do kwestii replikowalności wyników otrzymanych przez zespół. Tutaj są następujące uwagi i wyniki z naszej strony:





Wyniki ROC AUC na danych dostępnych tylko dla walidacji są bardzo zbliżone do tych co są stworzone przez grupę budującą. Czyli modele spełniają test walidacyjny i dalsze wyciągnięte z poprawności wybory modelu są poprawne.