Klasyfikacja płci głosu

Maciej Chylak, Maciej Gryszkiewicz

Kwiecień 2021

Contents

1 Wstep

Nasz projekt opiera sie na danych ze strony `https://apispreadsheets.com/datasets/119`. Owy zbiór danych składa sie z 3168 rekordów opisujacych różne charakterystyki głosu ludzkiego (np. czestotliwość, ton dominujacy itd.). Celem projektu jest utworzenie modelu klasyfikacyjnego, który jak najdokładniej bedzie przewidywał czy dany głos należy do kobiety, czy do meżczyzny.

2 EDA - badania eksploracyjne

Na poczatku badań eksploracyjnych zgłebiliśmy temat reprezentowania dźwieku w postaci liczb, w szczególności głosu ludzkiego. Pozwoliło to nam to lepiej zrozumieć dane, na których operowaliśmy. Oprócz tego, stworzyliśmy skrypt w R, którego zadaniem jest transformowanie nagrań głosu ludzkiego na ramki danych analogiczne do tych dostepnych na stronie apispreadsheets.com. Dzieki temu zabiegowi mogliśmy na koniec przetestować nasz model na samodzielnie nagranych próbkach.
Dalsza cześć EDA stanowiła wizualizacja danych. Zapoznaliśmy sie z rozkładami każdej ze zmiennej przy pomocy histogramów. Nastepnie zbadaliśmy korelacje zmiennych przy pomocy macierzy korelacji reprezentowanej przez heatmape. W kolejnym kroku sporzadziliśmy pairploty dla czterech najciekawszych zmiennych. Wybraliśmy te zmienne ze wzgledu na to, że różnica w dystrybuantach meskich i żeńskich była w ich wypadku najwieksza. Na koniec tego rozdziału stworzyliśmy 2 ridgeploty ilustrujace zmienne, których nie udało nam sie odtworzyć przy pomocy wyżej wspomnianego skryptu w R. Na nasze szczeście, zmienne te nie wnosiły zbyt wiele do naszego modelu, za ich pomoca nie dało rozróżnić sie głosu meskiego od żeńskiego.

3 Preprocessing

Poczatkowo przyjrzeliśmy sie rozkładowi zmiennych.W trakcie przegladania heatmapy zauważyliśmy, że kilka zmiennych, tj. centroid, kurt, dfrange jest mocno skorelowanych z innymi, wiec ostatecznie postanowiliśmy je wyrzucić. Nastepnie dokonaliśmy transformacji zmiennych skew, modindx oraz maxfun, gdyż wykresy tych zmiennych wykazywały duże skośności.

Zajeliśmy sie także analiza ciszy, gdyż zauważyliśmy, że wiekszość obserwacji zawiera bardzo długie fragmenty, w których do mikrofonu nic nie mówiono. Szczególnie widać to w przypadku zmiennej Q25, gdzie bardzo duża cześć naszych obserwacji posiada ta zmienna o wartości mniejszej niż 80Hz, a jak wiadomo głos ludzki rozpoczyna sie od około 80Hz. Postanowiliśmy zatem usunać wszystkie obserwacje, które maja wartość tej zmiennej poniżej 60Hz (Aby nie wyrzucić przypadkiem zbyt wielu obserwacji).

Sprawdziliśmy także, czy któraś z naszych zmiennych ma dystrybucje rozkładu normalnego. Zrobiliśmy to w celu łatwej możliwości usuniecia outlierów. W tym celu użyliśmy biblioteki scipy, lecz jak sie okazało żadna z tych zmiennych nie wykazała tej cechy (było to spowodowane tym, że test ten w sposób bardzo surowy podchodzi do definicji bycia rozkładem normalnym).

Ostatnia cześć preprocessingu stanowił encoding wartości label. W naszym wypadku 0 - oznaczać bedzie głos żeński, natomiast 1 - meski.

4 Wybór modelu

Przetestowaliśmy nastepujace modele:

- SVM
- KNNeighbors
- Regresja logistyczna
- Naiwny Klasyfikator Bayesowski
- Drzewo decyzyjne
- Model stackowany (DecisionTree, NB, LogisticRegression $->$ LogisticRegression)
- XGBoost

Głównym kryterium wyboru modelu był wynik "accuracy". Zdecydowaliśmy sie na użycie tej metryki, ponieważ nasz zbiór danych był idealnie zrównoważony. Do tego, z naszego punktu widzenia nie ma różnicy pomiedzy predykcja typu fals-positive, a false-negative. Oprócz tego, jest to miara najbardziej intuicyjna. Dodatkowo wspomagaliśmy sie metryka ROC AUC. Używaliśmy jej jako ogólny i uniwersalny wyznacznik jakości modelu, ale "accuracy" pozostała decydujacym czynnikiem.
Najlepiej wypadł model XGBoost i to jego zdecydowaliśmy sie użyć w dalszej cześci projektu. Bardzo dobrym kandydatem wydawał sie również model Decision Tree. Jego "accuracy" stało na bardzo przyzwoitym poziomie, a dodatkowo był w pełni przejrzysty i jawny.

5 Strojenie modelu

Aby polepszyć jakość wybranego modelu, zdecydowaliśmy sie na dostrojenie jego hiperparametrów. W tym celu użyliśmy metody RandomizedSearchCV. Skupiliśmy sie na najistotniejszych parametrach takich jak:

- learning_rate
- gamma
- max_depth
- nround
- n_estimators
- min_child_weight

Strojenie to nie zmieniło w istotnym stopniu jakości naszego modelu, niemniej jednak delikatnie poprawiło osiagane przez model "accuracy".

6 Test modelu na własnych danych

Na samym końcu postanowiliśmy sprawdzić, czy nasz model bedzie również działał w przypadku nagranych przez nas dźwieków. Do przetworzenia dźwieku posłużyliśmy sie wcześniej utworzona przez nas funkcja. Zamienia ona dźwiek z formatu wav na ramke danych, zawierajaca (prawie) te same zmienne co te zawarte na stronie `https://apispreadsheets.com/datasets/119`. Na nasze nieszczeście, model źle przewiduje ostateczne wyniki. Postanowiliśmy wiec dokonać dekompozycji predykcji dla naszych obserwacji i zauważyliśmy, że zmienna która najbardziej definiowała ostateczny wynik była zmienna meanfun, która w przypadku utworzonej przez nas ramki danych miała wartości nieadekwatne do tych, które były w przypadku pierwotnej ramki. To samo tyczy sie wszystkich zmiennych dotyczacych czestotliowści fundamentalnej oraz dominujacej. Być może jest to kwestia sprzetu nagrywajacego lub złego sposobu obliczania wartości tych zmiennych.