Zadania na 3.0 – sprawozdanie

Wszystkie zadania będą wykonywane przeze mnie przy użyciu Pythona i jego bibliotek. Komentarze w kodzie będę wykonywał w języku angielskim.

1. Stworzenie drzewa decyzyjnego na datasecie titanic (dostępny pod linkiem: <https://www.kaggle.com/competitions/titanic>)

Do wykonania tego zadania użyję biblioteki Scikit-Learn, do wczytania i manipulacji danymi będę używał Pandas.

Otwieram zestaw danych, do treningu i do testu, dane są w postaci plików CSV

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Interesuje mnie zestaw do treningu, z którego muszę wydzielić dane oraz target, którym będzie kolumna Survived.

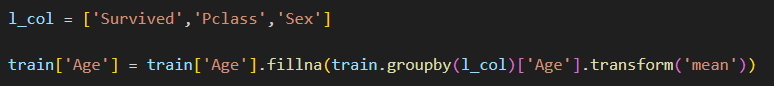
Zanim jednak to zrobie, warto byłoby uzupełnić brakujące dane wiekowe niektórych pasażerów. Wiek w tym przypadku pełni dosyć ważną rolę jeśli rozważamy przetrwanie pasażerów. Pustych danych jest całkiem sporo, co mogę łatwo sprawdzić sumując je:





Puste dane postanowiłem uzupełnić, grupując pasażerów kolumnami ['Survived','Pclass','Sex'] i wyciągając średnią wiekową z tych grup.

Uzupełniam dane przy użyciu .fillna():



Przed wrzuceniem danych do klasyfikatora, trzeba je jeszcze nieco wyczyścić oraz zamienić wartości słowne na liczbowe. Dodaje nową kolumnę FamilySize oraz IsAlone, rezygnując z niepotrzebnych.

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Przy użyciu biblioteki Seaborn, można w fajny sposób zwizualizować korelacje między danymi:

Obraz zawierający zrzut ekranu, tekst, kwadrat, Wielobarwność

Opis wygenerowany automatycznie

Chciałbym teraz sprawdzić, jak głębokie drzewo będzie najbardziej optymalne w naszym przypadku. Do tego, użyję sobie techniki Cross Validation, która będzie dzielić nasz train set na części i obliczać celność dla każdej z nich.

Możliwe głębokości ustawiłem na od 1 do ilości kategorii w datasecie, ilość podziałów na 10:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, czarne, Czcionka

Opis wygenerowany automatycznie

Widzę, że najlepiej poszło na 5, także wykorzystam to jako max głębokość swojego drzewa.

Robię sobie podział na dane i target:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Tworzę drzewo, wrzucam do niego dane, robię test na testowych danych.

Wytrenowany model zapisuje sobie, dodatkowo zapisuje sobie png z moim drzewem w celu wizualizacji

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Do stworzenia obrazu użyłem biblioteki Graphviz

Obraz drzewa zapisany jest w pliku tree\_image (za szeroki by tu wkleić).

Na koniec liczę sobie celność modelu, która wyniosła 86.08

1. Prosta sieć dwuwarstwowa ucząca się XORA
2. Sieć konwolucyjna ucząca się MNIST