234073	234102	234106
Numer indeksu	*Numer indeksu*	*Numer indeksu*
Aleksandra Kowalczyk	Zbigniew Nowacki	Karol Podlewski
Imię i nazwisko	*Imię i nazwisko*	*Imię i nazwisko*

Kierunek Informatyka Stosowana
Stopień II
Specjalizacja Data Science
Semestr 2

Data oddania 4 listopada 2020

Przetwarzanie i analiza dużych zbiorów danych

Zadanie 2

Spis treści

1 Cel zadania

Celem zadania była implementacja algorytmu "Osoby, które możesz znać", który jest wykorzystywany przede wszystkim przez media społecznościowe do sugerowania nowych, potencjalnych znajomych.

Każdy użytkownik posiadający znajomych dostaje rekomendacje do 10 nowych użytkowników, którzy nie są jego znajomymi, a mają najwięcej wspólnych znajomych. W przypadku użytkowników nie posiadających znajomych, program powinien zwrócić pustą listę.

2 Opis implementacji

Do implementacji zadania wykorzystano język Python wraz z API PySpark, które umożliwiło skorzystanie z możliwości języka Apache Spark w pythonowym kodzie. Działanie programu można przedstawić w następujących krokach:

1. Z podanego pliku tekstowego wczytano dane zawierające identyfikatory użytkowników oraz identyfikatory ich znajomych
2. Dla każdego użytkownika stworzono listę rekomendacji dla jego znajomych łącząc wszystkich ze sobą w pary - następnie usunięto każde wystąpienie pary znajomych, którzy już są znajomymi, po czym obliczono jak często dana para występuje
3. Dla każdego użytkownika przypisano parę w postaci nowego sugerowanego znajomego oraz liczby wspólnych znajomych, którą następnie posortowano po liczbie wspólnych znajomych i identyfikatorze użytkownika
4. Do tej listy dodano użytkowników, dla których algorytm nie wyznaczył żadnych rekomendacji

2.1 Wykorzystanie Apache Sparka

W celu implementacji algorytmu wykorzystano kolekcje RDD (Resilient Distributed Datasets), będące podstawową kolekcją obiektów Sparka, które są przystosowane do pracy z BigData. Obiekty RDD były poddawane przeróżnym transformacjom takim jak filtracja czy mapowanie. Niezbędne okazało się także korzystanie z redukcji czy sortowania obiektów po kluczu - niezwykle przydatna okazała się redukcja po kluczu `reduceByKey()`, która znacznie ułatwiła proces zliczania par o identycznych kluczach.

3 Uzyskane wyniki

W tabeli poniżej przedstawiono rekomendacje uzyskane dla użytkowników o identyfikatorach: 924, 8941, 8942, 9019, 9020, 9021, 9022, 9990, 9992 oraz 9993.

Tabela 1: Użytkownicy wskazani w poleceniu oraz proponowani im znajomi

Użytkownik	Rekomendacje
924	439, 2409, 6995, 11860, 15416, 43748, 45881
8941	8943, 8944, 8940
8942	8939, 8940, 8943, 8944
9019	9022, 317, 9023
9020	9021, 9016, 9017, 9022, 317, 9023
9021	9020, 9016, 9017, 9022, 317, 9023
9022	9019, 9020, 9021, 317, 9016, 9017, 9023
9990	13134, 13478, 13877, 34299, 34485, 34642, 37941
9992	9987, 9989, 35667, 9991
9993	9991, 13134, 13478, 13877, 34299, 34485, 34642, 37941