**SSIED Projekt**

Pump it Up

Martin Czakański

Adam Krawiec

Kuba Łada

Michał Slabik

## Tomasz Ścigała

# **Wprowadzenie**

Dane zostały dostarczone przez Taarifa oraz Wodne Ministerstwo w Tanzani. Celem projektu jest przewidzenie stanu pomp wodnych na terenie kraju w Tanzani. Pompy wodne mogą być w jednym z trzech następujących stanów:

* Functional - sprawna
* Functional needs repair - sprawna ale wymaga naprawy
* Non Functional - nie sprawna

Projekt jest realizowany na potrzeby przedmiotu Sieci Społeczne i Eksploracja Danych.  
Konkurs jest darmowy i każdy może wziąć w nim udział. Dodatkowo za najlepsze rozwiązanie problemu nie ma żadnych przewidzianych nagród.

Więcej informacji na temat konukrsy można znaleźć pod poniższym linkiem: <https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table>

# **Opis zawartości plików źródłowych**

W konkursie są udostępniane czterych pliku źródłowe w formacie .csv. Pliki **SubmissionFormat.csv** oraz **Training set values.csv** zawierają etykiety dla zbioru testowego oraz dla pliku do odesłania.

# SubmissionFormat.csv oraz Training set values.csv

* + - Id – Id pompy wodnej
    - status\_group - stan pompy wodnej możliwe wartości:
      * functional
      * non functional
      * functional needs repair

Pliki **Test set values.csv** oraz **Training set values.csv** zawierają dane testowe oraz dane do przeprowadzenia analizy.

Test set values.csv oraz Training set values.csv

* Id – identyfikator pompy wodnej
* amount\_tsh - ilośc dostępnej wody
* date\_recorded - data wprowadzenia danych
* funder - fundator pompy wodnej
* gps\_height - wysokość geograficzna
* installer - organizacja odpowiedzialna za instalacje
* longitude - długość geograficzna
* latitude - szerokość geograficzna
* wpt\_name - nazwa
* num\_private - ?
* basin - zbiornik wodny
* subvillage - lokalizacja - wioska
* region - lokalizacja - region
* region\_code - lokalizacja - kod regionu
* district\_code - lokalizacja - kod dzielnicy
* lga - lokalizacja - ?
* ward - lokalizacja - oddział
* population – liczba populacji
* public\_meeting - czy jest to w miejscu publicznym
* recorded\_by – organizacja odpowiedzialna za wpisany rekord
* scheme\_management - kto jest operatorem
* scheme\_name – nazwa operatora
* permit – czy punkt wodny jest dozwolony
* construction\_year – rok produkcji
* extraction\_type – rodzaj wydobycia wody
* extraction\_type\_group – grupa wydobycia wody
* extraction\_type\_class – klasa wydobycia wody
* management – nazwa zarządcy
* management\_group – nazwa grupy zarządzającej
* payment – koszt wody
* payment\_type – typ płatności
* water\_quality – jakość wody
* quality\_group – grupa jakości wody
* quantity – ilośc wody
* quantity\_group – grupa ilości wody
* source - źródło
* source\_type – typ źródła
* source\_class – klasa źródła
* waterpoint\_type – typ pompy wodnej
* waterpoint\_type\_group – grupa typu pompy wodnej

# **Oczyszczanie danych**

# Usunięcie nadmiarowych kolumn oraz uzupełnienie brakujących danych

Dane zostały na początku załadowane i wyświetlone w programie RapidMiner, następnie na \ zaczeliśmy robić analize wszystkich kolumn usuwając nadmiarowe kolumny oraz ustawialiśmy domyślne dane, jeśli w którejś kolumnie brakowały zawartości.

Poniższa tabela przedstawia jakie opreracje zostały wykonywane na każdej kolumnie:

|  |  |  |
| --- | --- | --- |
| Nazwa kolumny | Kolumna została usunięta | Uzupełniono brakujące wartości |
| id | ✓ | 🗶 |
| amount\_tsh | 🗶 | 🗶 |
| date\_recorded | 🗶 | 🗶 |
| funder | 🗶 | ✓ |
| gps\_height | ✓ | 🗶 |
| installer | 🗶 | ✓ |
| longitude | ✓ | 🗶 |
| latitude | ✓ | 🗶 |
| wpt\_name | ✓ | 🗶 |
| num\_private | ✓ | 🗶 |
| basin | ✓ | 🗶 |
| subvillage | ✓ | 🗶 |
| region | ✓ | 🗶 |
| region\_code | ✓ | 🗶 |
| district\_code | ✓ | 🗶 |
| lga | ✓ | 🗶 |
| ward | ✓ | 🗶 |
| population | 🗶 | 🗶 |
| public\_meeting | 🗶 | ✓ |
| recorded\_by | ✓ | 🗶 |
| scheme\_management | 🗶 | ✓ |
| scheme\_name | ✓ | 🗶 |
| permit | 🗶 | ✓ |
| construction\_year | 🗶 | 🗶 |
| extraction\_type | ✓ | 🗶 |
| extraction\_type\_group | ✓ | 🗶 |
| extraction\_type\_class | ✓ | 🗶 |
| management | ✓ | 🗶 |
| management\_group | ✓ | 🗶 |
| payment | ✓ | 🗶 |
| payment\_type | 🗶 | 🗶 |
| water\_quality | 🗶 | 🗶 |
| quality\_group | ✓ | 🗶 |
| quantity | 🗶 | 🗶 |
| quantity\_group | ✓ | 🗶 |
| source | ✓ | 🗶 |
| source\_type | 🗶 | 🗶 |
| source\_class | 🗶 | 🗶 |
| waterpoint\_type | 🗶 | 🗶 |
| waterpoint\_type\_group | 🗶 | 🗶 |

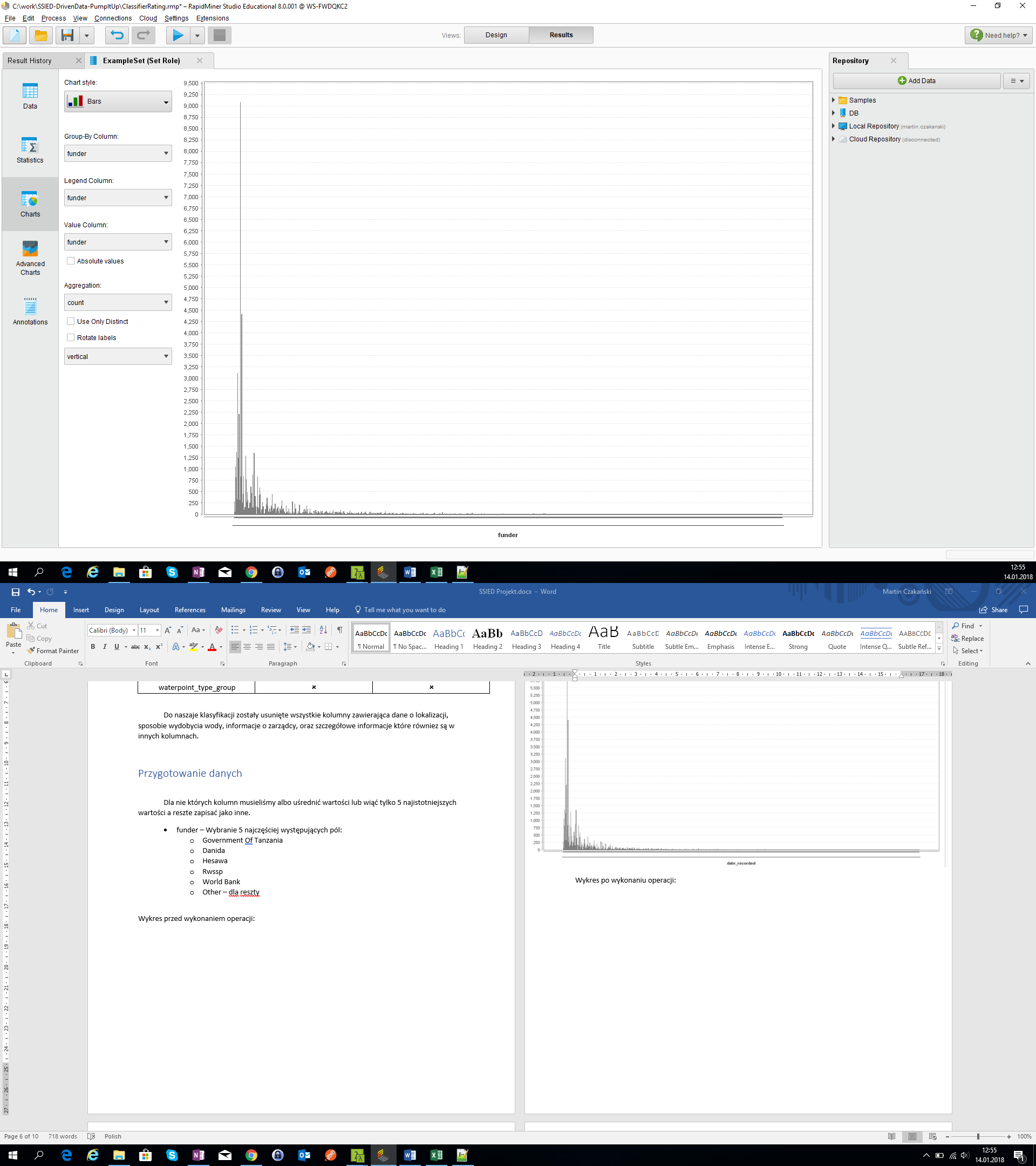
Do naszaje klasyfikacji zostały usunięte wszystkie kolumny zawierająca dane o lokalizacji, sposobie wydobycia wody, informacje o zarządcy, oraz szczegółowe informacje które równiez są w innych kolumnach.

# Przygotowanie danych

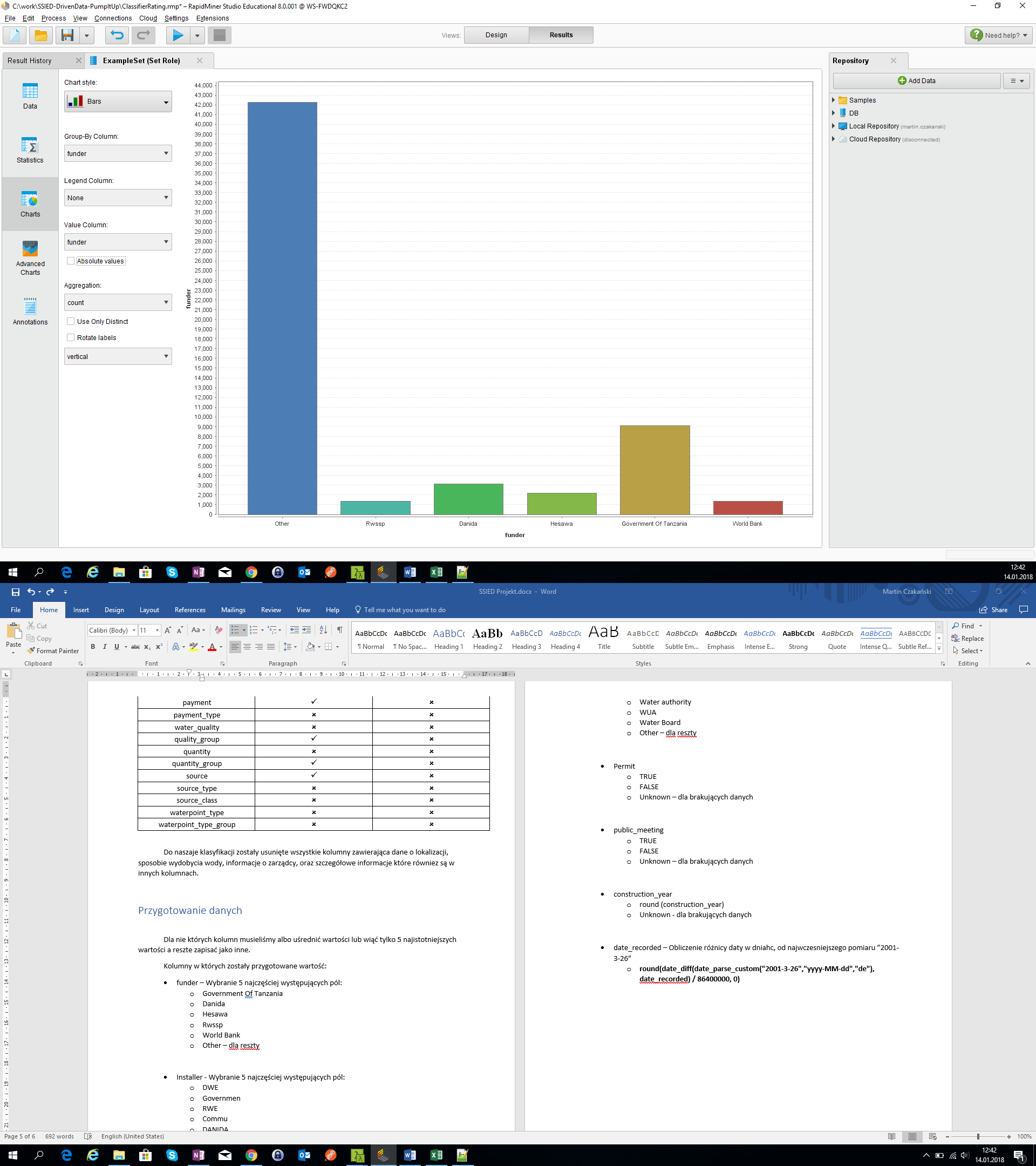
Dla nie których kolumn musieliśmy albo uśrednić wartości lub wiąć tylko 5 najistotniejszych wartości a reszte zapisać jako inne.

* funder – Wybranie 5 najczęściej występujących pól:
  + Government Of Tanzania
  + Danida
  + Hesawa
  + Rwssp
  + World Bank
  + Other – dla reszty

Wykres przed wykonaniem operacji:

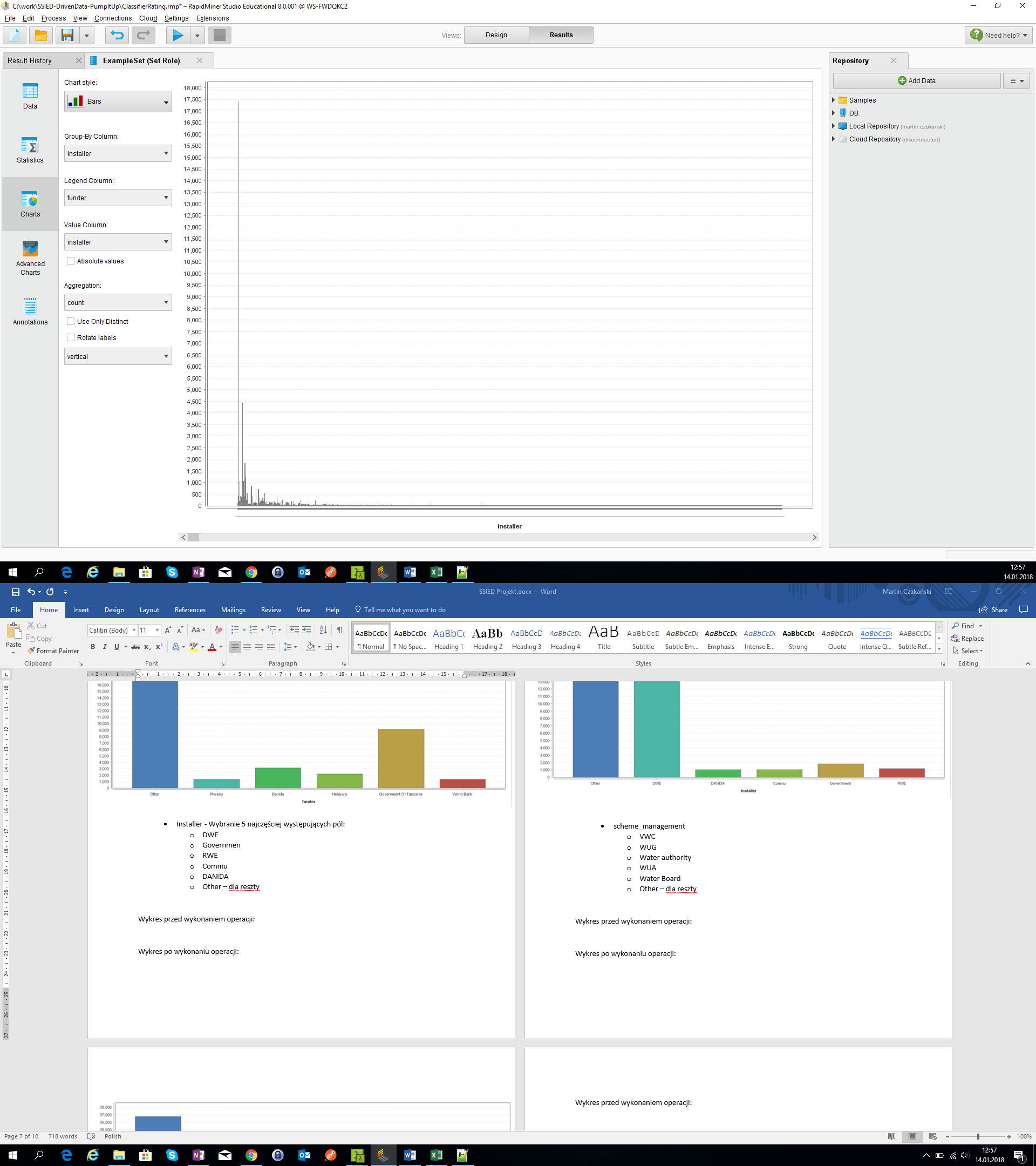


Wykres po wykonaniu operacji:

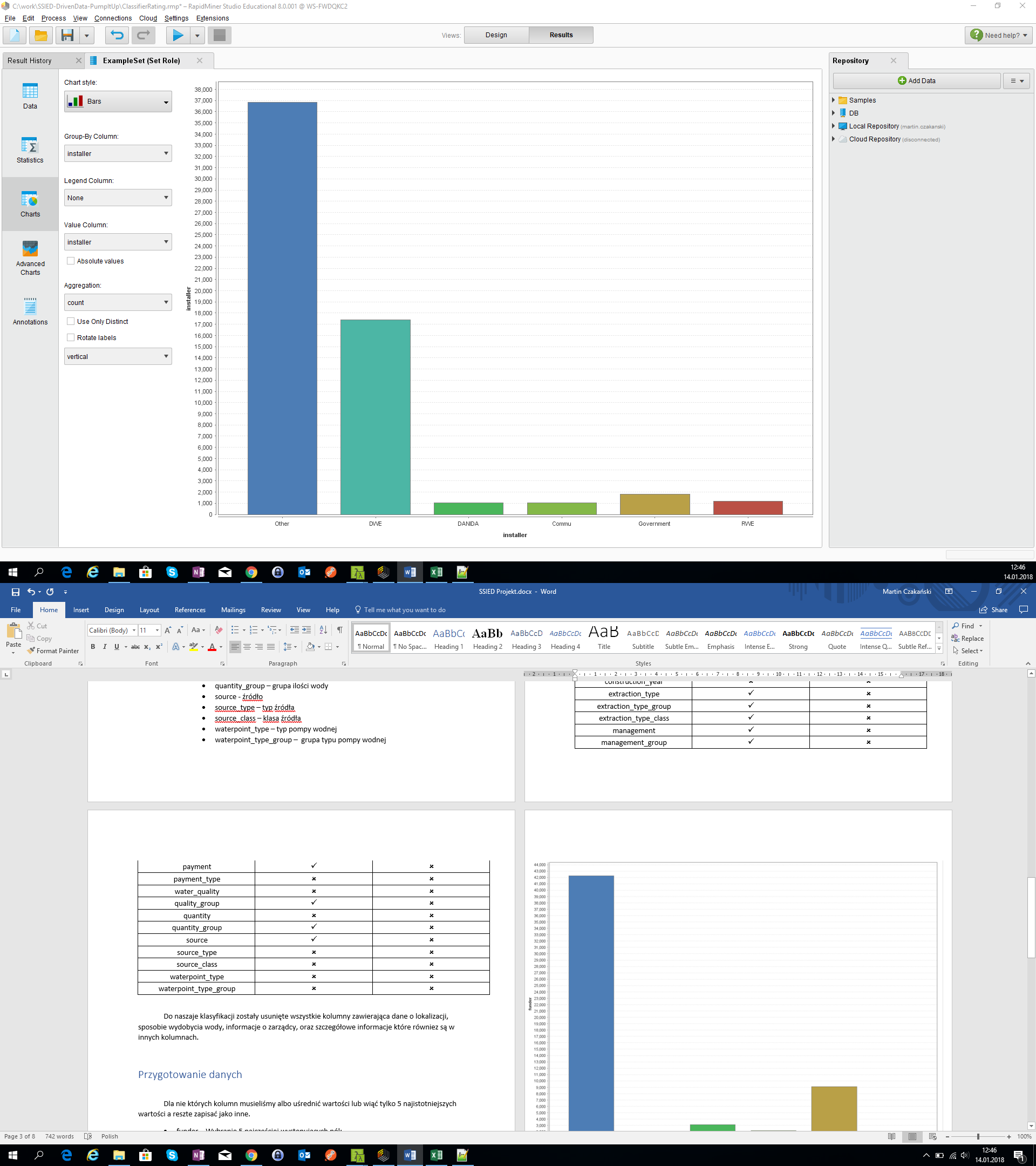


* Installer - Wybranie 5 najczęściej występujących pól:
  + DWE
  + Governmen
  + RWE
  + Commu
  + DANIDA
  + Other – dla reszty

Wykres przed wykonaniem operacji:

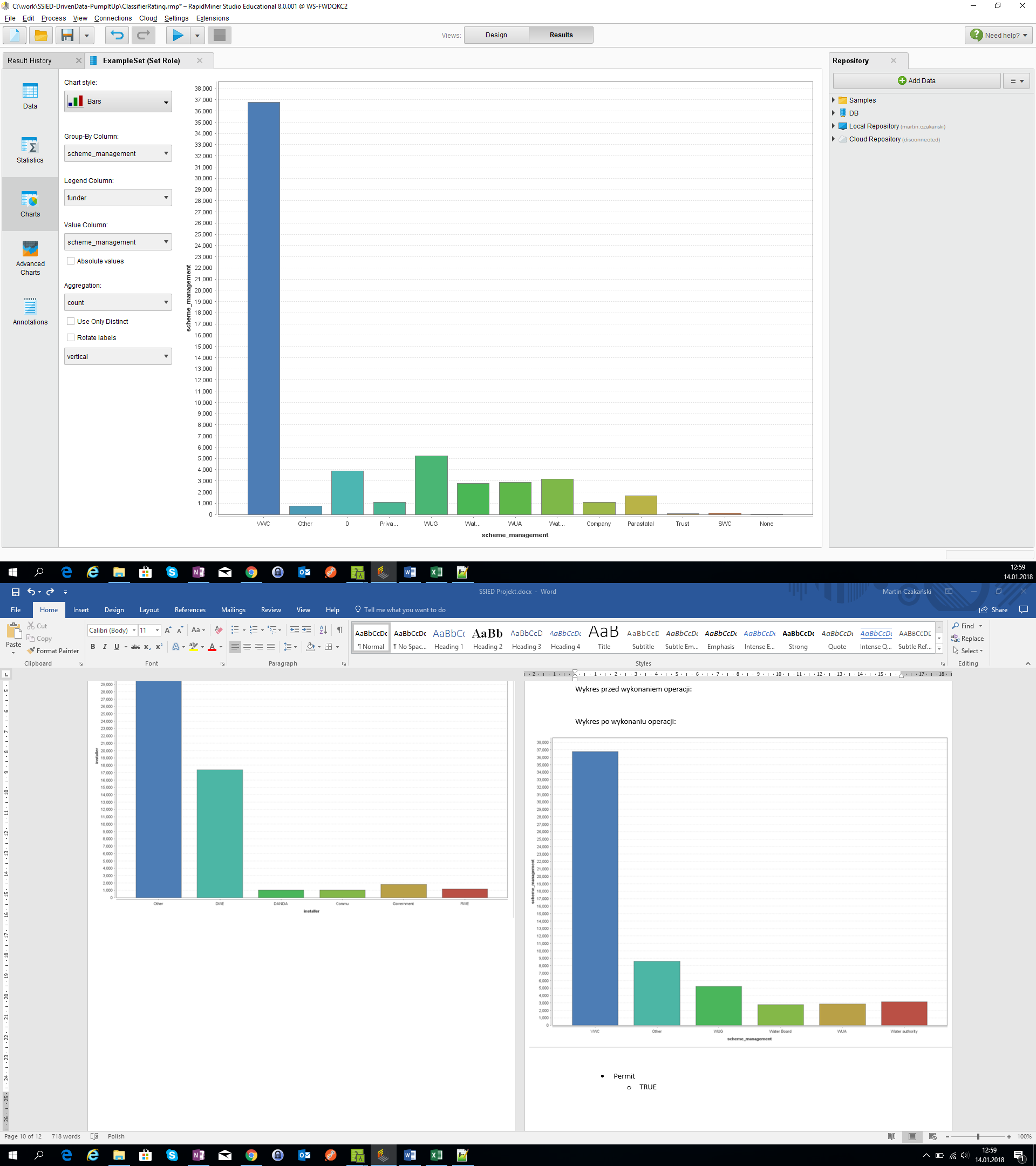


Wykres po wykonaniu operacji:

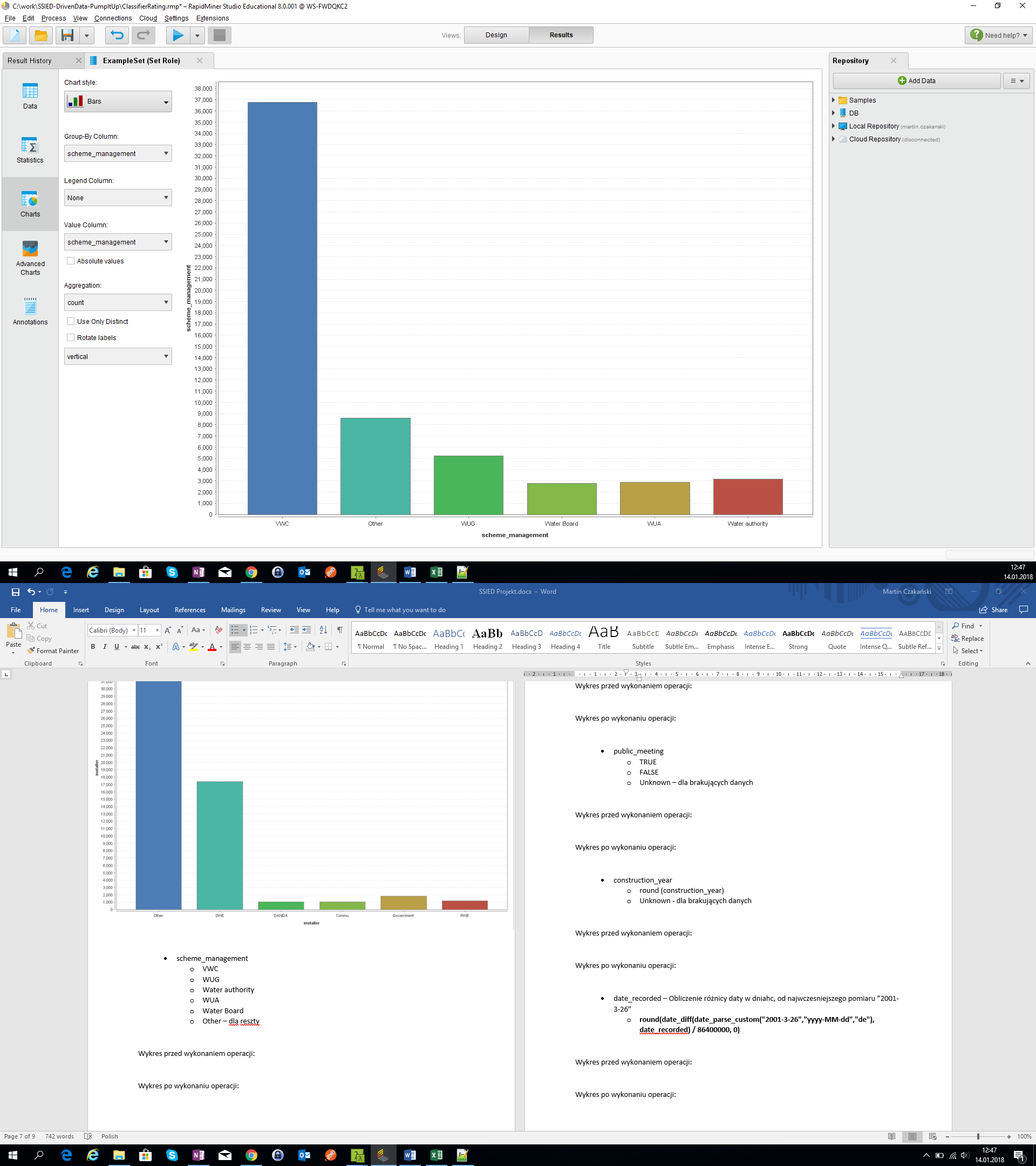


* scheme\_management
  + VWC
  + WUG
  + Water authority
  + WUA
  + Water Board
  + Other – dla reszty

Wykres przed wykonaniem operacji:

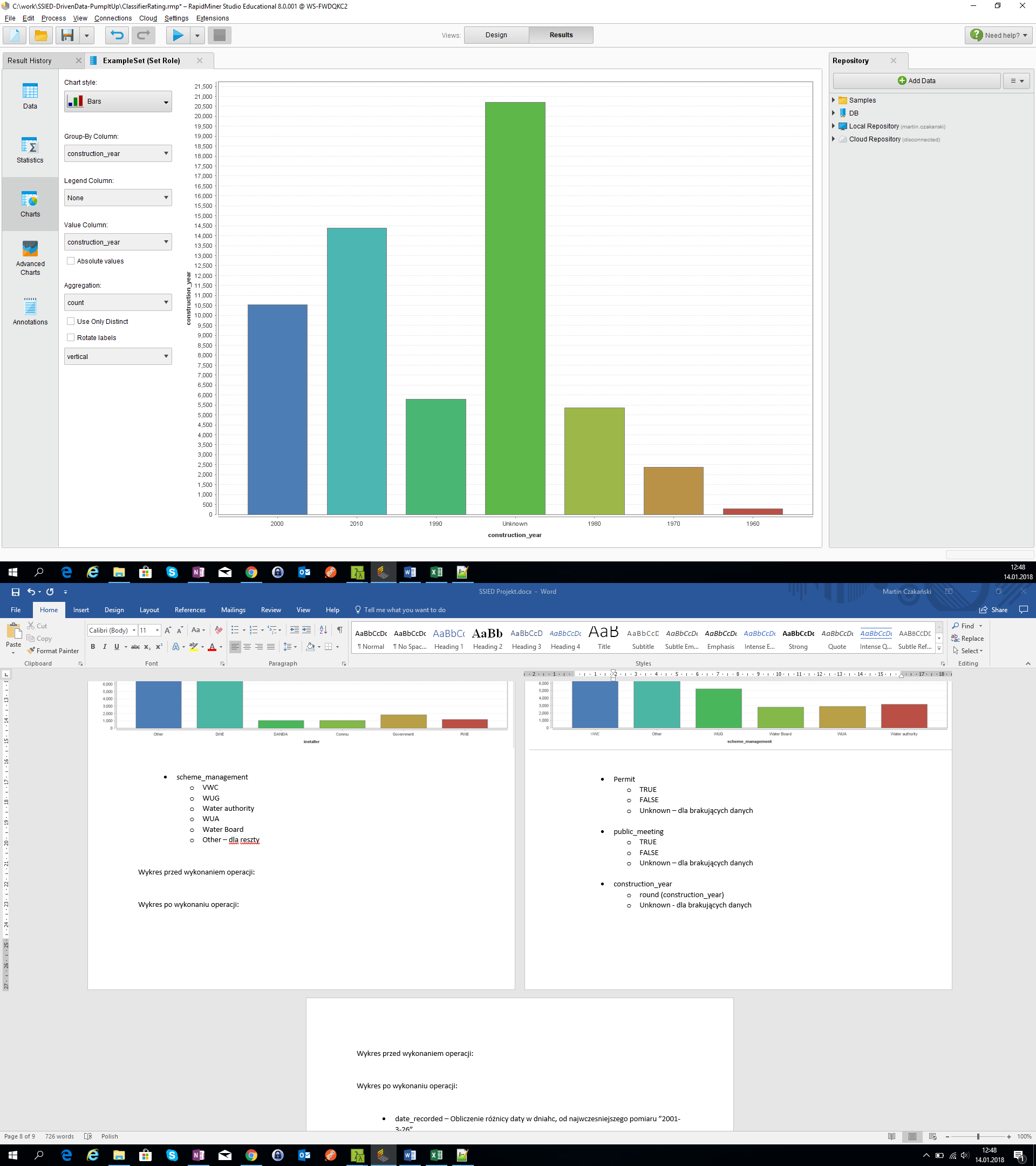


Wykres po wykonaniu operacji:



* Permit
  + TRUE
  + FALSE
  + Unknown – dla brakujących danych
* public\_meeting
  + TRUE
  + FALSE
  + Unknown – dla brakujących danych
* construction\_year
  + round (construction\_year)
  + Unknown - dla brakujących danych

Wykres po wykonaniu operacji:

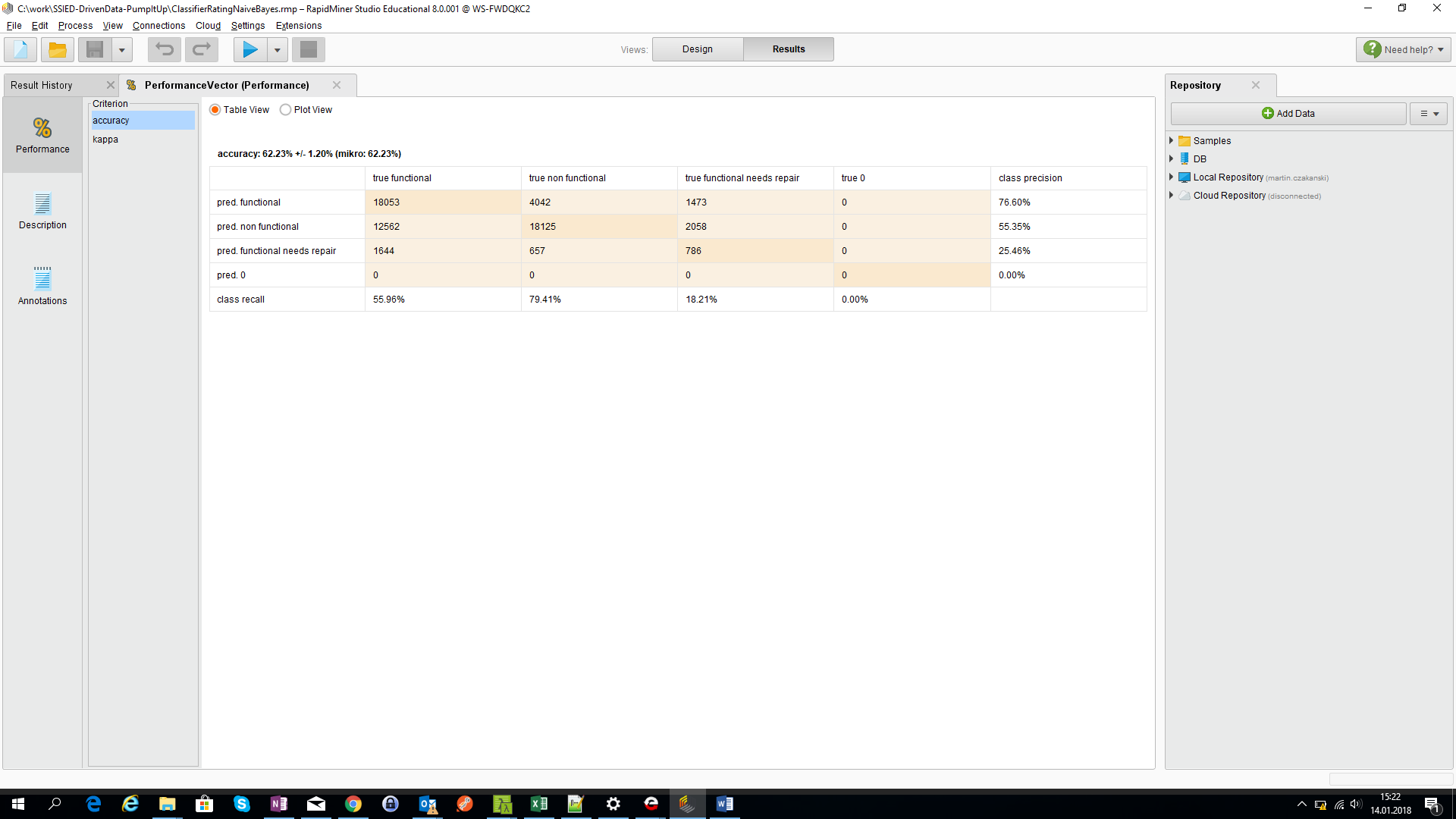


* date\_recorded – Obliczenie różnicy daty w dniach, od najwczesniejszego pomiaru ”2001-3-26”
  + **round(date\_diff(date\_parse\_custom("2001-3-26","yyyy-MM-dd","de"), date\_recorded) / 86400000, 0)**

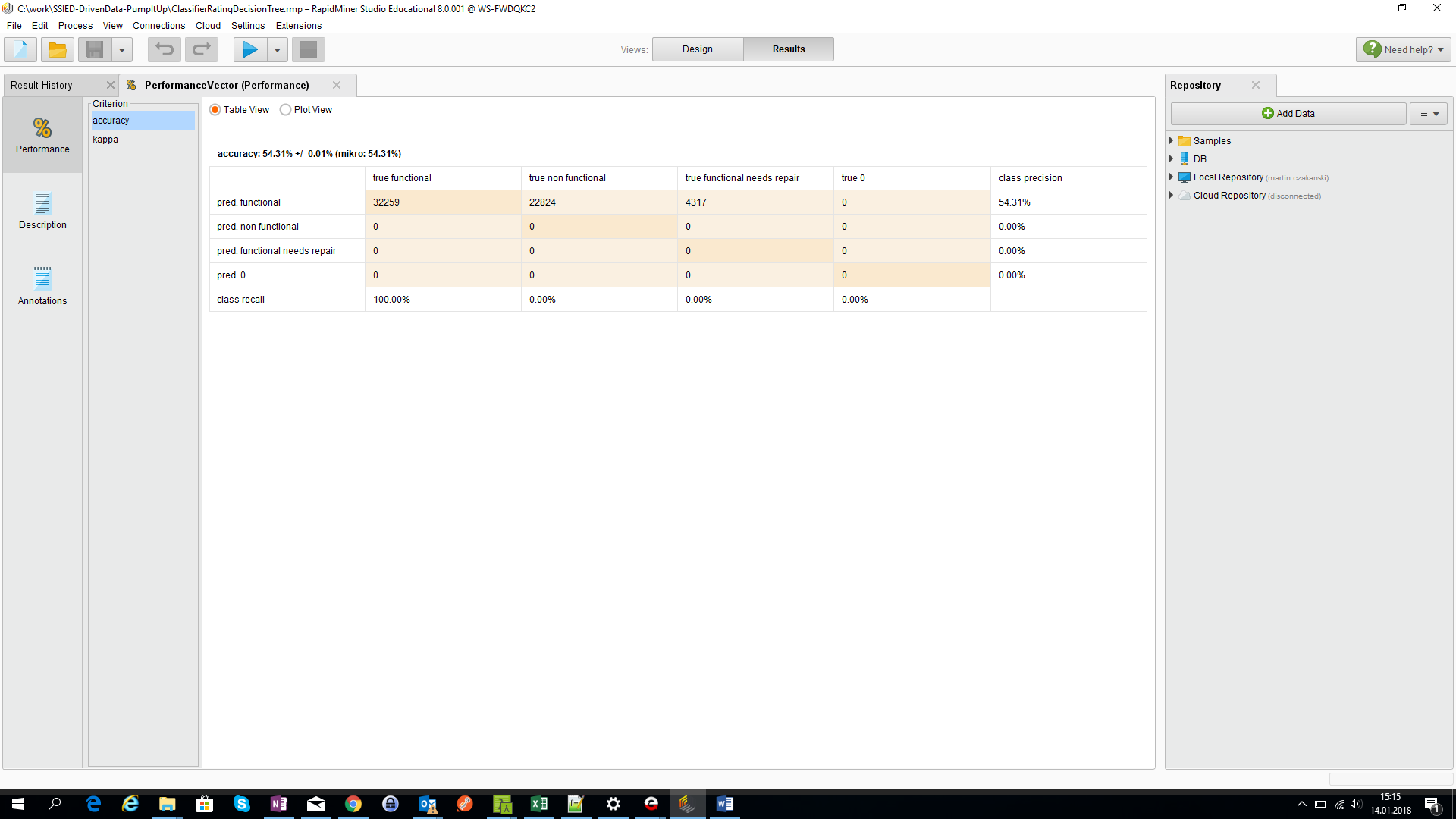
# **Klasyfikacja danych**

Do klasyfikacji danych przetestowaliśmy 3 różne algorytmy. Najlepiej wywapadł algorytm Gradient Boosted Trees i z wyników tego algorytmu skorzystaliśmy do przesłania wyników konkursowych.

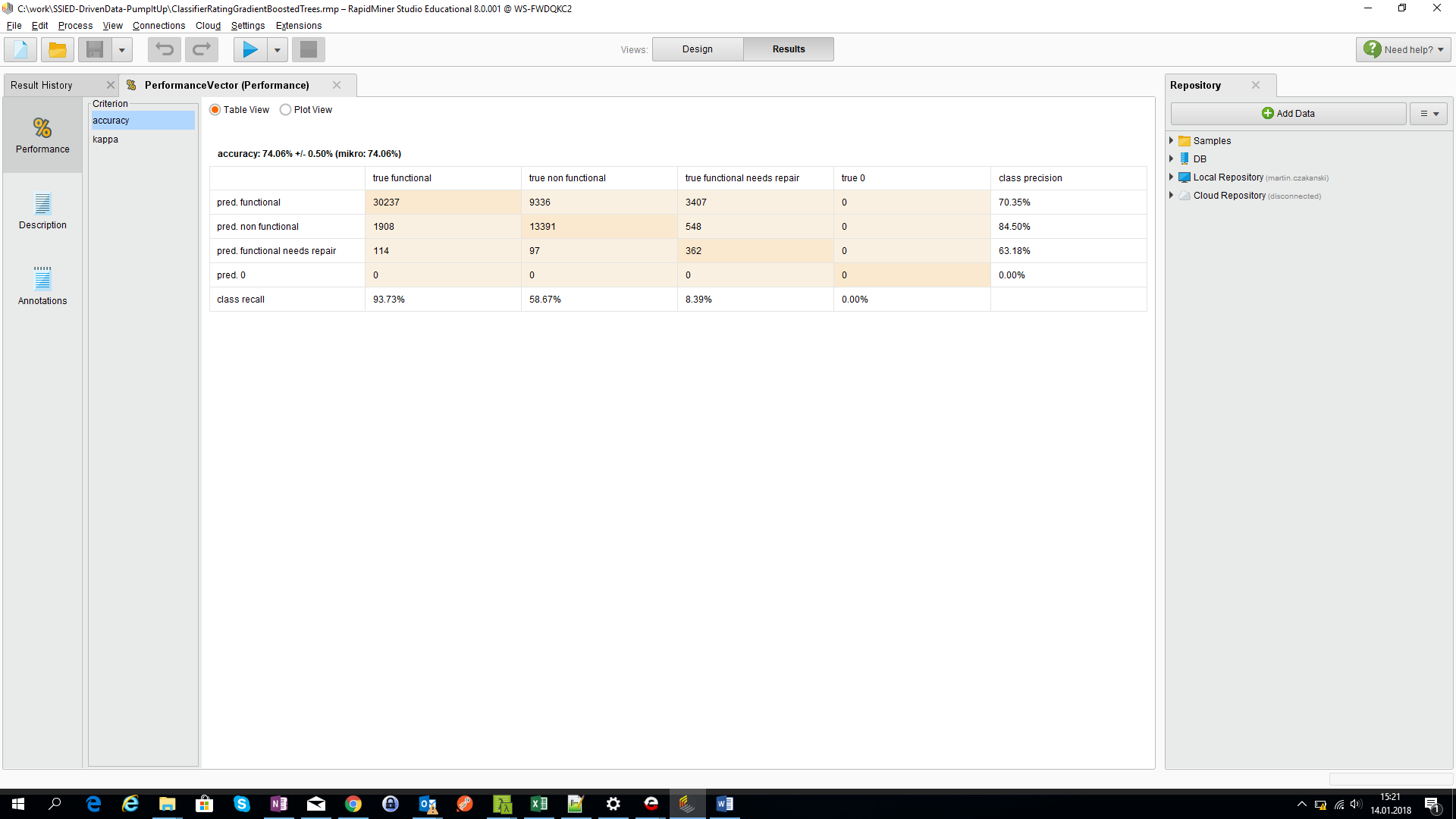
* Naive Bayes



* Decision Tree

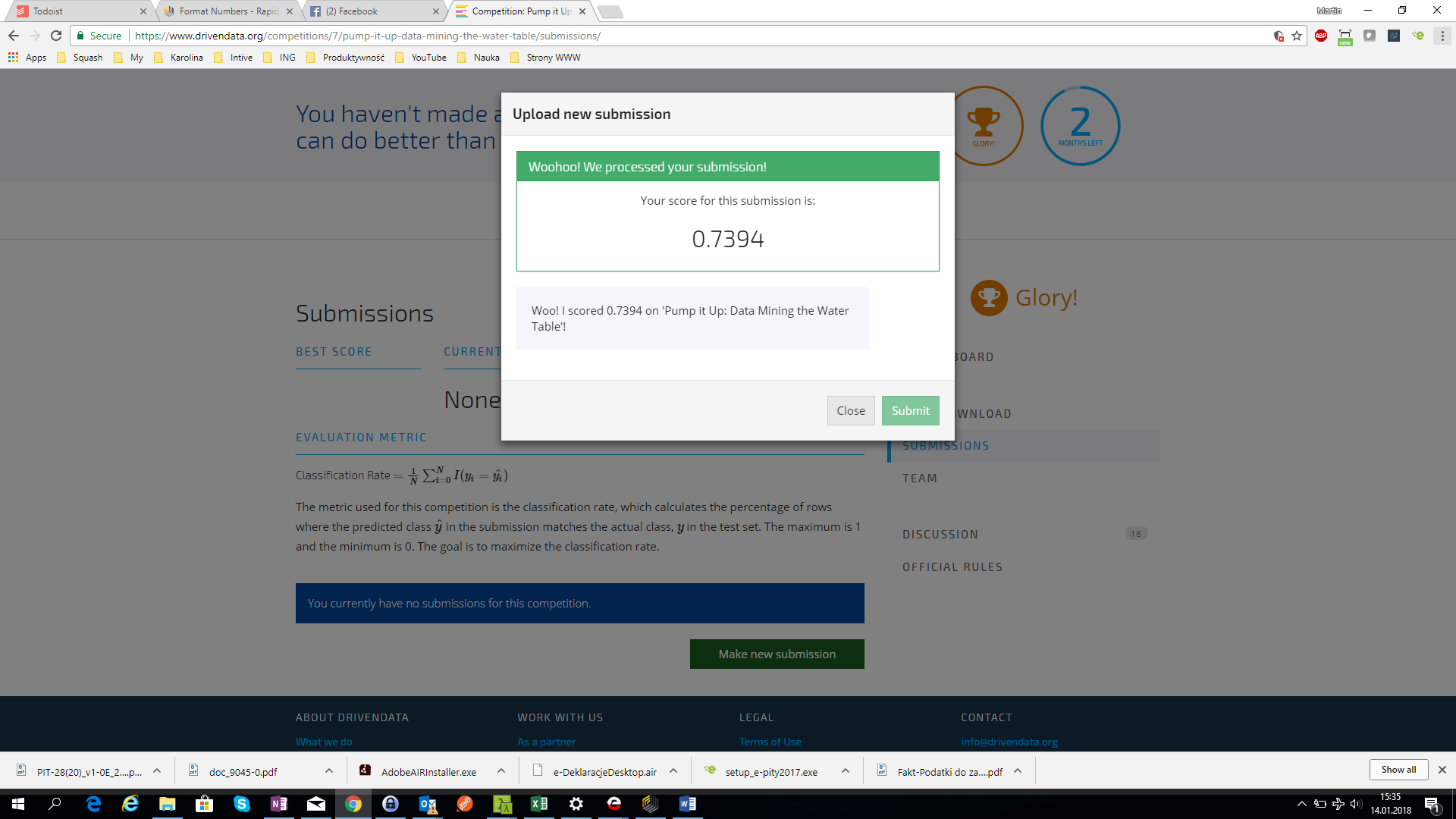


* Gradient Boosted Trees



# **Wynik konkursowy**

Po wysłaniu danych otrzymaliśmy poniższy wynik:



# **Wnioski**

* Oprogramowanie RapidMiner bardzo ułatwia analizowanie dnaych i wykorzystywanie gotowych komponentów. Program jest bardzo intuicyjny i w szybkim czasie można wykonać analize na danych. Dodatkowo dzięki gotowym komponentów, likwidujemy problemy składniowe, które mogą się pojawiąc podczas wykorzystania takich języków jak R lub python.
* Dane zawierają bardzo wiele informacji na różnych stopniach szczegółowości, dzięki temu możemu wybrać jak bardzo dokładnie chcemy zbudować nasz klasyfikator.
* Dzięki dużemu zbiórowi dostępnych algorymtów, w bardzo szybki sposób (zmieniająć „bloczek”), jesteśmy w stanie sprawdzić inny algorytm klasyfikacji danych. Najlepszy wynik otrzymaliśmy korzystając z algorytmu Gradient Boosted Trees