**Monika Szymańska 238359**

Zadanie 4

# 4.1 analiza tekstu w RapidMiner

Korzystając z procesu Processing Text and Document Objects oraz przykładowych danych JobPosts (w moim przypadku zapisanych jako Zadanie\_4.1d) dokonałam analizy danych tekstowych w podprocesach:

* Po wstępnej tokenizacji
* Po redukcji stopwords i odrzuceniu tokenów według długości
* Po utworzeniu n-gramów
* Po zastosowaniu stemmingu

Fragment oryginalnego zbioru danych oraz otrzymanych przekształceń prezentuje Tabela 1.

|  |  |
| --- | --- |
| **Oryginalny tekst** | Level 2 Safety Office for Burnaby construction site (residential/multi-family project). Start date - immediate. Only those with level 2 need apply. |
| **Po wstępnej tokenizacji** | level safety office for burnaby construction site residential multi family project start date immediate only those with level need apply |
| **Po redukcji stopwords i odrzuceniu tokenów według długości** | level safety office burnaby construction site residential multi family project start date immediate level apply |
| **Po utworzeniu n-gramów** | level level\_safety level\_safety\_office safety safety\_office safety\_office\_burnaby office office\_burnaby office\_burnaby\_construction burnaby burnaby\_construction burnaby\_construction\_site construction construction\_site construction\_site\_residential site site\_residential site\_residential\_multi residential residential\_multi residential\_multi\_family multi multi\_family multi\_family\_project family family\_project family\_project\_start project project\_start project\_start\_date start start\_date start\_date\_immediate date date\_immediate date\_immediate\_level immediate immediate\_level immediate\_level\_apply level level\_apply apply |
| **Po zastosowaniu stemmingu** | level safeti offic burnabi construct site residenti multi famili project start date immedi level appli |

Tabela - Fragment dokumentu (15) po przeprowadzeniu procesu Processing Text and Document Objects

Jak widzimy otrzymany tekst po wstępnej tokenizacji jest pomniejszony o wszystkie elementy, które nie są słowami czyli np. znaki interpunkcyjne czy liczby. Oryginalny tekst po redukcji stopwords i odrzuceniu tokenów według długości znacznie nam się zmniejszył ponieważ z tekstu zostały usunięte słowa, które nie wpływają bezpośrednio na treść i znaczenie tekstu jak np. „for”. Po utworzeniu n-gramów widzimy, że otrzymaliśmy zestawienie wyrazów, kolejno występujących po sobie w tekście (2 słowa i 3 słowa ponieważ n = 3), natomiast kiedy został użyty stemming widzimy, że słowa zostały przekształcone do formy podstawowej, czyli do tzw. korzenia wyrazu.

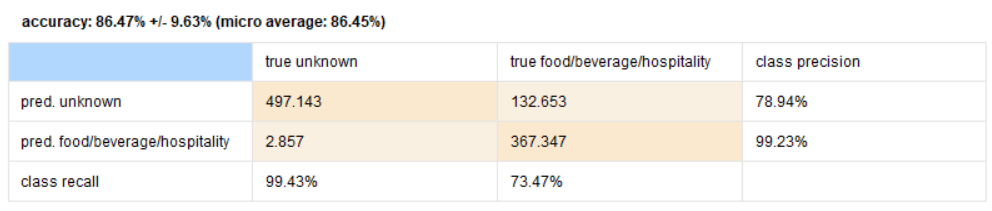
Analiza pod kątem najczęściej występujących słów dla danych podprocesach została przedstawiona w Tabeli 2.

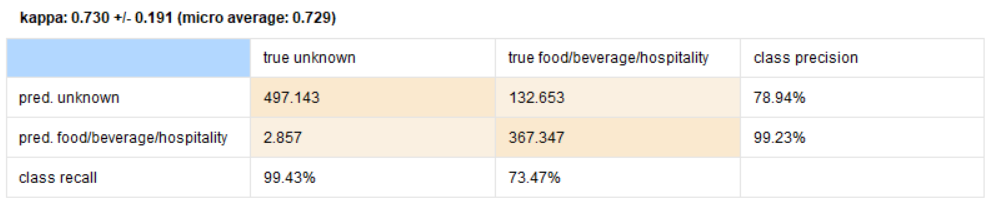
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Podproces** | **Po wstępnej tokenizacji** | | **Po redukcji stopwords i odrzuceniu tokenów według długości** | | | **Po utworzeniu n-gramów** | | | **Po zastosowaniu stemmingu** | |
| **słowo** | **Liczba wystąpień** | **słowo** | **Liczba wystąpień** | **słowo** | | **Liczba wystąpień** | **słowo** | | **Liczba wystąpień** |
| **1.** | and | 3200 | experience | 566 | experience | | 566 | work | | 688 |
| **2.** | to | 2213 | work | 471 | work | | 471 | experi | | 579 |
| **3.** | a | 1790 | please | 394 | please | | 394 | pleas | | 396 |
| **4.** | the | 1701 | skills | 307 | skills | | 307 | requir | | 353 |
| **5.** | in | 1225 | looking | 281 | looking | | 281 | skill | | 338 |
| **6.** | for | 1149 | team | 271 | team | | 271 | posit | | 325 |
| **7.** | of | 1109 | time | 256 | time | | 256 | look | | 313 |
| **8.** | with | 999 | resume | 247 | resume | | 247 | time | | 312 |
| **9.** | you | 797 | position | 246 | position | | 246 | servic | | 300 |
| **10.** | is | 723 | sales | 246 | sales | | 246 | custom | | 292 |

Tabela – Top 10 najczęściej występujących słów w danych podprocesach

Analizując otrzymane wyniki pod względem najczęściej występujących termów widzimy, że po zastosowaniu tylko wstępnej tokenizacji przeważa liczba słów, które są spójnikami, rodzajnikami czy słowami pomocniczym. Kiedy już zostały one odrzucone (podproces: po redukcji stopwrds i odrzuceniu tokenów według długości) widzimy, że najczęściej występującym słowem jest „experience”, które pojawiło się w tekście 566 razy, natomiast dodanie n-gramów nie wpłynęło na liczbę występujących słów w tekście. Po zastosowaniu stemmingu widzimy, że najczęściej występującą postacią bazową jest „work”, które dla dwóch poprzednich przypadków plasowało się na drugiej pozycji. Różnica wystąpienia cząstki „work”, a słowa „work” wynosi 217, a zatem tyle razy w tekście wystąpiło słowo, które zawierało w sobie ten wyraz.

Dla procesu Applying a Model Categorize Documents i przykładowych danych JobPosts została wykonana klasyfikacja dokumentów tekstowych zapisanych w plikach html. Pliki zostały sklasyfikowane na podstawie zawierających termów. Wyniki dla zbioru JobPosts zostały przedstawione na Rysunku 1.





Rysunek - Wyniki klasyfikacji dla zbioru danych JobPosts

Jak widać powyżej dokładność klasyfikacji dla zbioru JobPosts wyniosła 86,47%, natomiast wskaźnik Kappa wyniósł 0,730. Zatem możemy stwierdzić, że klasyfikacja jest na przeciętnym poziomie, jednak w wyższej jego granicy (Kappa > 0,75 – klasyfikacja doskonała).

Poniżej przedstawiono wyniki klasyfikacji dla zbioru testowego.





Rysunek - Wyniki klasyfikacji dla zbioru testowego

Jak można zauważyć na Rysunku 2 jakoś klasyfikacji różni się w stosunku do zbioru JobPosts. Dokładność osiągnęła 93%, natomiast wartość wskaźnika Kappa wyniosła 0,596. Na podstawie Kappy możemy stwierdzić, że wynik klasyfikacji jest średni pomimo tego, że procentowa dokładność klasyfikacji jest wysoka.

Dokonano zmiany sposobu generowania termów z 1 na 4 wyrazy, aby sprawdzić czy ma to pozytywny wpływ na wynik klasyfikacji. Wyniki tego porównania zostały przedstawione w Tabeli 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **JobPosts** | | **Testing set** | |
| **n-grams** | 1 | **4** | **1** | **4** |
| **Accuracy** | 86,47% | 78,38% | 93,00% | 93,00% |
| **Kappa** | 0,730 | 0,569 | 0,596 | 0,557 |

Tabela - Wyniki klasyfikacji po zmianie n-gram

Zmiana n-gram z 1 na 4 spowodowała, że dla zbioru danych JobPosts poprawność klasyfikacji oraz wskaźnik Kappa zmalał. Dla zbioru testowego nie odnotowano zmian dokładności klasyfikacji, natomiast Kappa zmalała z 0,596 do 0,557. Można zatem stwierdzić, że zwiększenie wartości n-gram wpływa negatywnie na jakość klasyfikacji.

W Tabeli 4 przedstawiono 10 najczęściej występujących słów dla n-gram równego 1 oraz 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N - gram** | **1 - gram** | | **4 - gram** | |
| **słowo** | **Liczba wystąpień** | **słowo** | **Liczba wystąpień** |
| **1.** | work | 527 | work | 527 |
| **2.** | experi | 492 | experi | 492 |
| **3.** | pleas | 323 | pleas | 323 |
| **4.** | requir | 294 | requir | 294 |
| **5.** | skill | 269 | skill | 269 |
| **6.** | posit | 267 | posit | 267 |
| **7.** | look | 251 | look | 251 |
| **8.** | time | 242 | time | 242 |
| **9.** | custom | 232 | custom | 232 |
| **10.** | servic | 232 | servic | 232 |

Tabela - Top 10 najczęściej występujących słów po zmianie n-gram

Jak widać powyżej zmiana wartości n-gram nie ma wpływu na częstotliwość występowania danych słów w tekście.

W bloku Cross Validation został zamieniony klasyfikator SVM na Decission Tree (przy wartości n-gram równej 1). Otrzymane wyniki zostały przedstawione w Tabeli 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **JobPosts** | | **Testing set** | |
| **klasyfikator** | **SVM** | **Decission Tree** | **SVM** | **Decision Tree** |
| **Accuracy** | 86,47% | 82,39% | 93,00% | 89,00% |
| **Kappa** | 0,730 | 0,642 | 0,596 | 0,497 |

Tabela - Porównanie wyników klasyfikacji ze względu na użyte klasyfikatory

Zmiana klasyfikatora z SVM na Decission Tree sprawiła, że zarówno jakość klasyfikacji jak i wskaźnik Kappa pogorszyły się.

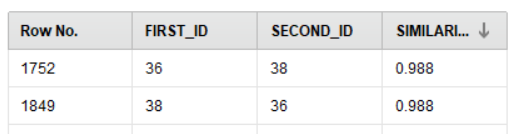
W tym samym procesie została zmieniona liczba n-gram z 1 do 4 aby sprawdzić czy przy zastosowaniu klasyfikatora Decission Tree zachodzi taka sama zależność jak dla SVM. Porównanie wyników zostało przedstawione w Tabeli 6.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **JobPosts** | | **Testing set** | |
| **n-grams** | 1 | **4** | **1** | **4** |
| **Accuracy** | 82,39% | 79,37% | 89,00% | 94,00% |
| **Kappa** | 0,642 | 0,582 | 0,497 | 0,694 |

Tabela - Zmiana wartości n-gram dla klasyfikatora Decission Tree

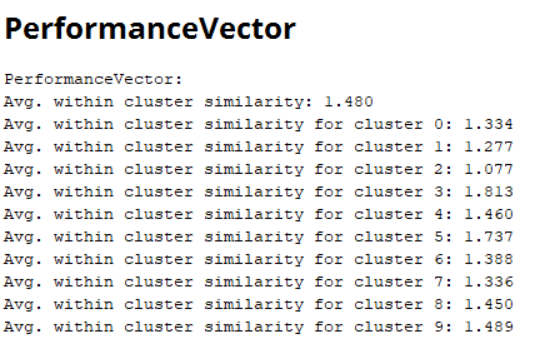
Dla danych JobPosts zmiana wartości n-gram z 1 na 4 sprawiła, że poprawność klasyfikacji jak i Kappa zmniejszyła się, zatem odnotowano pogorszenie się klasyfikacji. Odwrotny wynik natomiast możemy zaobserwować dla zbioru testowego, ponieważ w tym przypadku procentowa poprawność klasyfikacji wzrosła o 5%, a wskaźnik Kappa o 0,197.

Dla procesu Document Similarity and Clustering oraz danych JobPosts przeprowadzono proces działania operatorów analizy podobieństwa dokumentów. Analizując otrzymane wyniki możemy stwierdzić, że największe wartość miary podobieństwa występują w przypadku dokumentu nr 36 i 38 (Rysunek 3).



Rysunek - Fragment macierzy podobieństwa dla danych JobPosts

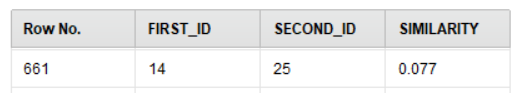
Badając zaś utworzone klastry (k=10) pod względem odległości wewnątrz skupień możemy stwierdzić, że najmniejsze wartości występują dla grupy 2, a następnie 1 (Rysunek 4), zatem dokumenty tam się znajdujące są najbardziej do siebie podobne.



Rysunek - Odległości od centrów skupień w utworzonych klastrach

Sprawdzając jakie cechy czyli w tym przypadku wyrażenia w tekście decydują o uznaniu ofert pracy znajdujących się w grupie 2 za podobne możemy wyróżnić tu m.in. bread (0,251), energetic (0,193) oraz intermediate (0,095). Dla grupy 1 możemy wyróżnić natomiast class (0,111), tonne (0,116) i drivers (0,067).

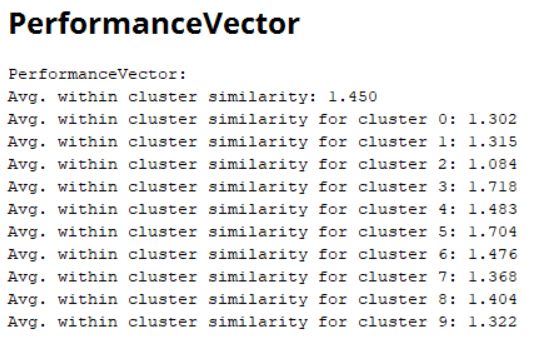
W klastrze 2 znajdują się pliki o numerach 14 i 25. Podobieństwo dla tych plików oczytana z macierzy podobieństwa została przedstawiona na Rysunku 5.



Rysunek - Podobieństwo między plikami z klastra 2 - 14 i 25

Można zatem zauważyć, że mimo tego że pliki zostały przydzielone do jednej grupy nie są one do siebie bardzo zbliżone ponieważ ich podobieństwo wynosi zaledwie 0,077.

Do bloku Proccess Documents form Data został dodany operator Generate n-grams aby sprawdzić czy wpływa to na jakość klasyfikacji. Wyniki przeprowadzonego procesu zostały zaprezentowane na Rysunku 6.

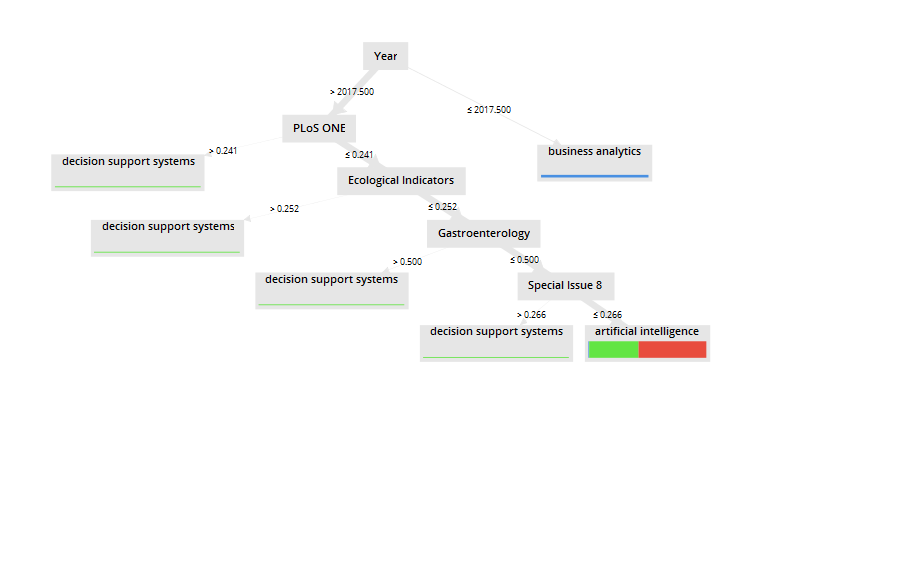


Rysunek - Odległość od centrów skupień w utworzonych klastrach po dodaniu operatora Generate n-gram

Jak widać powyżej średnia odległość wewnątrz skupień minimalnie zmalała (z 1,480 do 1,450), a zatem możemy stwierdzić, że dodanie operatora Generate n-gram pozytywnie wpłynęło na jakość grupowania.

Dokonano analizy danych tekstowych dla zbioru scopus, zwierającego dane bibliograficzne artykułów naukowych. Proces zawiera operatory Generae n-grams oraz Filter Token. Z otrzymanych wyników możemy zauważyć, że najczęściej występującym termem jest „Knowledge-Based Systems”, który pojawił się w tekście 15 razy. Bigramem, który pojawiał się w tekście najczęściej jest „Information Sciences\_451-452” (3 razy). Nie odnotowano natomiast trigramów, które pojawiają się więcej niż raz.

Proces został uzupełniony o grupowanie oraz klasyfikację za pomocą Decision Tree. Procentowa poprawność klasyfikacji dla zbioru danych scopus wyniosła 61,80%, natomiast wartość Kappa była równa 0,293, zatem stwierdza się klasyfikację niedostateczną. Otrzymane drzewo zostało zaprezentowane na Rysunku 7.



Rysunek - Drzewo klasyfikacyjne dla danych scopus

Jak widać powyżej najważniejszym atrybutem okazał się „Year”, a następnie „PLoS ONE” oraz „business analytics”, nie znalazły się w nim natomiast bigramy i trigramy. Nawet najczęściej występujący term „Knowledge-Based Systems” nie znalazł się w regułach decyzyjnych.

W wyniku grupowania plików otrzymano 10 klastrów (k=10). Najmniejsze to klastry 1 i 8 zawierające tylko 1 plik, natomiast największy to klaster 7 zawierający 278 dokumentów. Badając czy najczęściej występujący bigram ma wpływ na otrzymane grupy widzimy, że oddziałuje on tylko w niewielkim stopniu na klaster 0 (Rysunek 8).



Rysunek - Wpływ najczęściej występującego bigramu na utworzone grupy

W wyniku przebiegu procesu została utworzona macierz odległości, która została zapisana do pliku macierz\_zad4.1e.csv. Największe występujące odległości pomiędzy plikami wynoszą około 13,11 i odnotowano ich 474, natomiast najmniejsze wynoszą 0 i jest ich 94. Należy jednak uwzględnić, że pary występują podwójnie np. 86 i 84 oraz 84 i 86, zatem liczba ich jest o połowę mniejsza.

# 4.2 sentiment analysis w RapidMiner: kategoryzacja i grupowanie dokumentów

Korzystając z przykładowego procesu Sentiment Analysis dokonano analizy sentymentu na podstawie zbioru tekstów z nadanymi wartościami atrybutu sentiment. Po przeprowadzeniu klasyfikacji za pomocą klasyfikatora SVM otrzymaliśmy poprawność klasyfikacji na poziomie 63,50% oraz wskaźnik Kappa równy 0,244.

Wynik klasyfikacji dla zbioru testowego okazał się negatywny o wartościach confidence(negative) równej 0,587 oraz confidence(positive) wynoszącej 0,413.

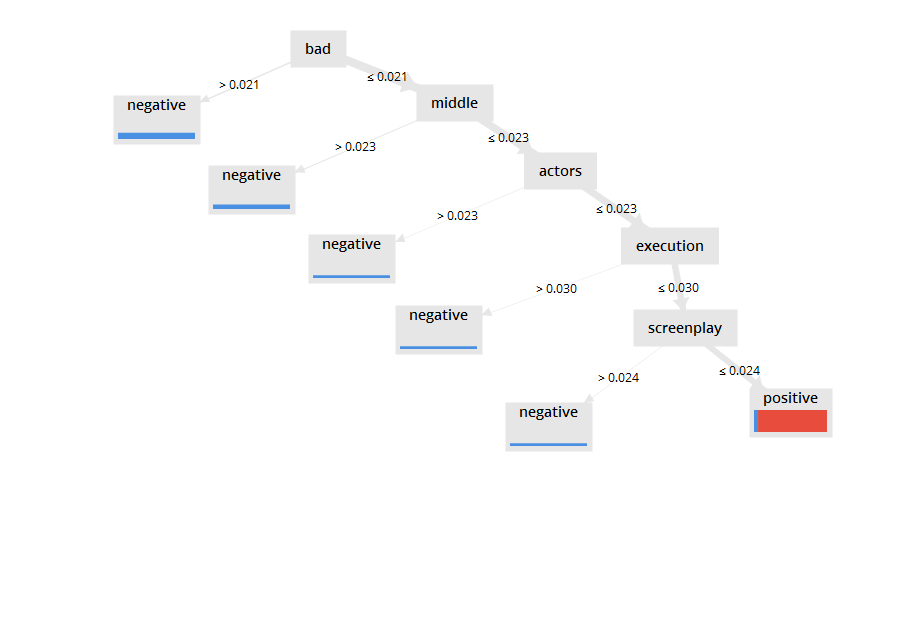
Dodanie bloku Weight by Information Gain pozwala sprawdzić jakie wyrażenia są najistotniejsze do stwierdzenia pozytywnego lub negatywnego charakteru analizowanego tekstu. Fragment otrzymanych wyników przedstawiono na Rysunku 9.



Rysunek - Najistotniejsze wyrażenia wykorzystane do określenia charakteru tekstu

Jak widzimy, najbardziej istotnym słowem, które wpływa na wydźwięk tekstu jest „bad”, następie „worst”. Wysoko w stawce znalazły się również takie słowa jak „house”, „saw” czy też „good”.

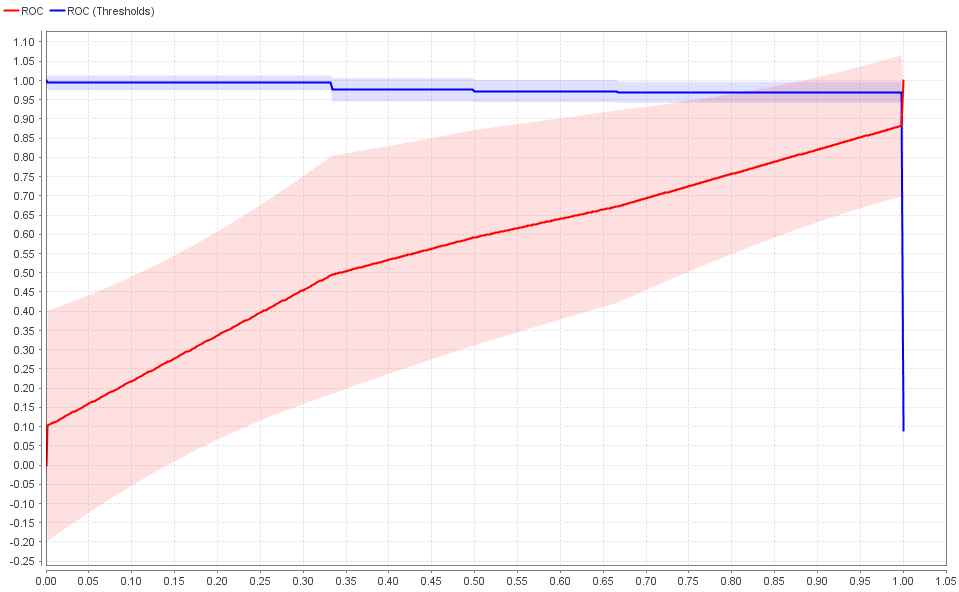
Klasyfikator SVM został zastąpiony Decision Tree w celu stwierdzenia czy wpływa to na jakość klasyfikacji. Otrzymane w wyniku przeprowadzenia procesu drzewo decyzyjne zostało zaprezentowane poniżej.



Rysunek - Drzewo decyzyjne otrzymane w wyniku procesu Sentiment Analysis

Na Rysunku 10 możemy zauważyć, że najbardziej istotnym atrybutem pozwalającym określić charakter tekstu jest „bad”. Jego wartość już na poziomie 0,021 pozwala uznać dokument za negatywny.

Wynik klasyfikacji po zmianie operatora plasuje się na poziomie 58,33% dla jakości klasyfikacji oraz 0,108 dla Kappa. Wykres przedstawiający krzywą ROC z przeprowadzonego procesu prezentuje Rysunek 11.



Rysunek - Krzywa ROC z przeprowadzonego procesu Sentiment Analysis z użyciem klasyfikatora Decision Tree

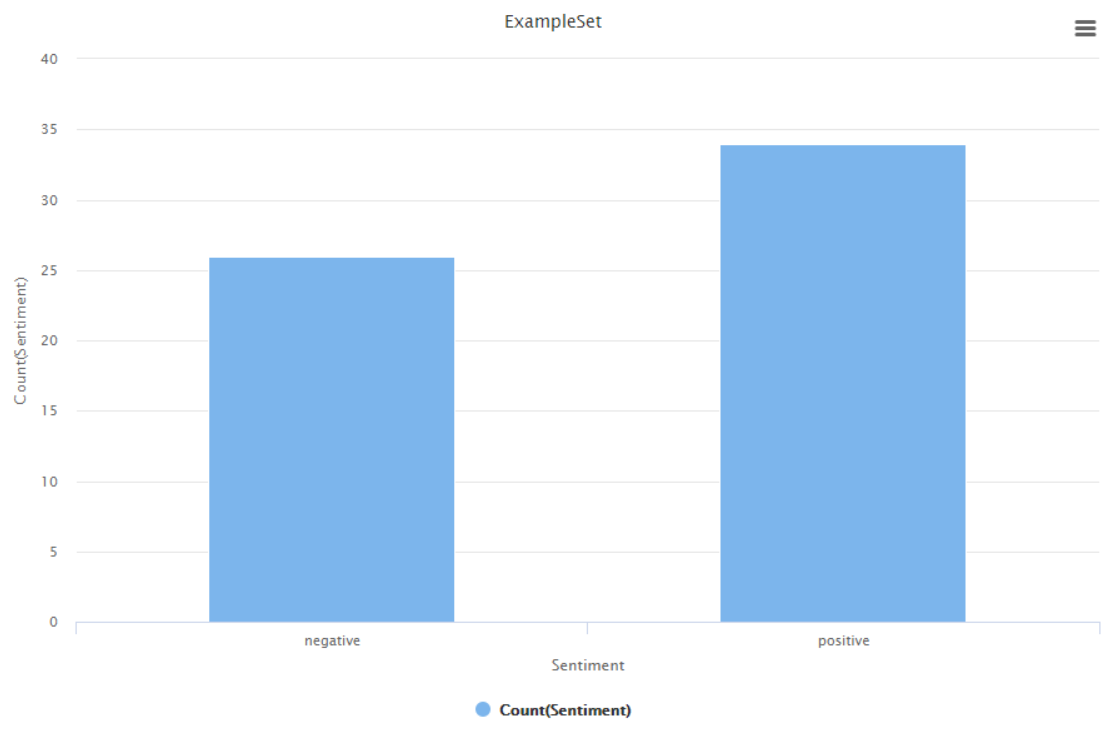
Można zatem stwierdzić, że zmiana klasyfikatora z SVM na Decision Tree wpłynęła negatywnie na jakość klasyfikacji. W obu przypadkach można było uznać klasyfikację za niedostateczną.

W kolejnym etapie wykorzystano operatory Extract Sentiment. Przeprowadzone analizy na zbiorze testowym z wykorzystaniem model sentiwordnet oraz vader dały nam następujące wyniki (Tabela 7).

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Prediction (Sentiment)** | **Confidence (negative)** | **Cnfidence (positive)** |
| sentiwordnet | positive | 0,056 | 0,944 |
| vader | positive | 0,056 | 0,944 |

Tabela - Wynik przeprowadzone procesu z wykorzystaniem Exract Seniment dla różnych modeli analizy tekstu

Aby sprawdzić jak zinterpretowane byłyby teksty ze zbioru uczącego po operatorze Cross Validation dodano blok Extract Sentiment. To jak rozkładały się dokumenty ze względu na ich wydźwięk prezentuje Rysunek 12.



Rysunek - Podział dokumentów ze względu na ich wydźwięk

Na powyższym wykresie widać, że większość tekstów na charakter pozytywny (34do 26).

Aby sprawdzić czy bardziej dokładna analiza tekstów poprawia wskaźniki klasyfikacji do bloku Process Documents from Data dodano operator Generate n-Grams. Procentowa poprawność klasyfikacji wówczas osiągnęła wynik 61,67%, natomiast wskaźnik Kappa 0,190, a zatem wynik w niewielkim stopniu się poprawił.

Korzystając z przykładu Sentiment Analysis dokonano oceny charakteru dokumentów znajdujących się w pliku twitter-iphone-100.csv. Przeprowadzając standardową analizę sentymentu z wykorzystaniem operatora Extract Sentiment uzyskaliśmy dla wszystkich plików wartość pozytywną (Rysunek 13).



Rysunek - Wynik oceny charakteru dokumentów znajdujących się w pliku twitter-iphone-100.csv