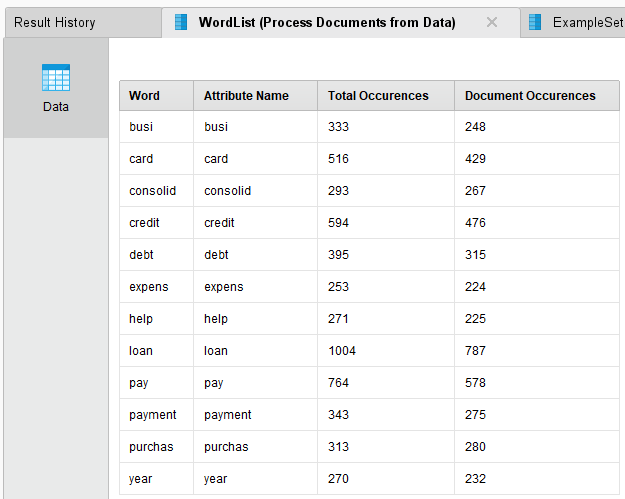
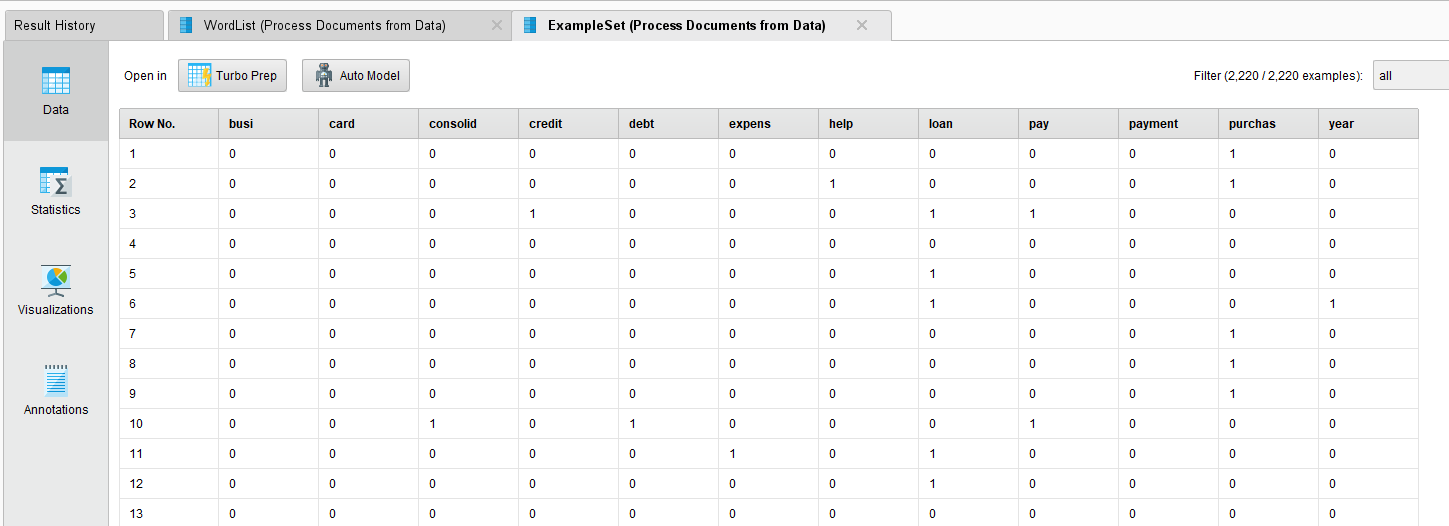
Process Documents from Data

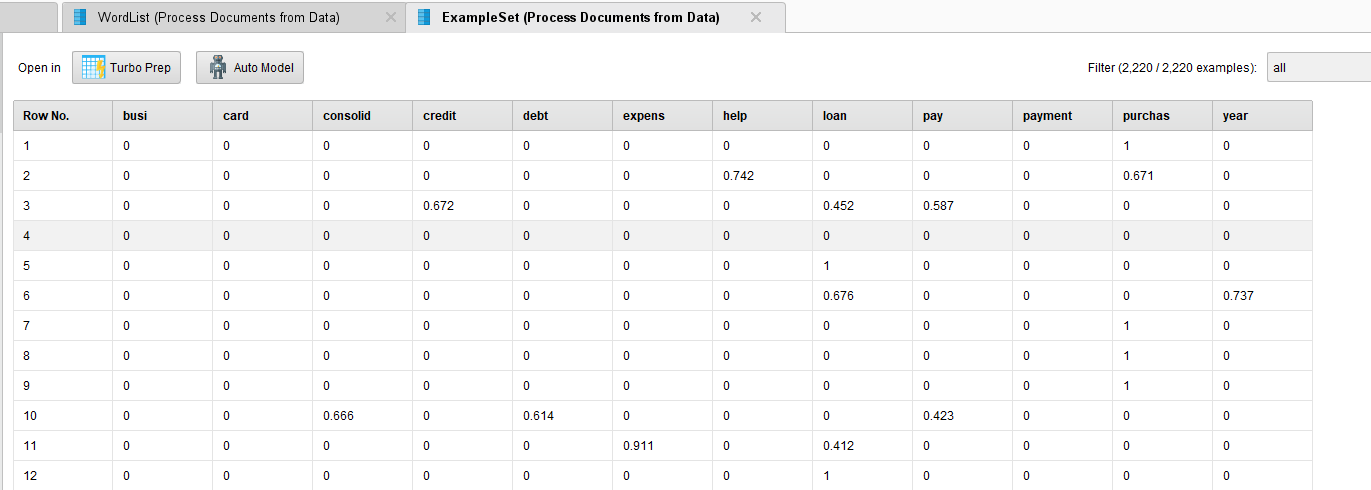
En el wordList resultante vemos el conglomerado de ocurrencias de las palabras relevantes encontradas en la columna de descripción. Para un total de 2220 documentos (registros) la palabra loan (préstamo) aparece en 787 documentos (registros) es la más ocurrente en el conjunto completo y por documento.



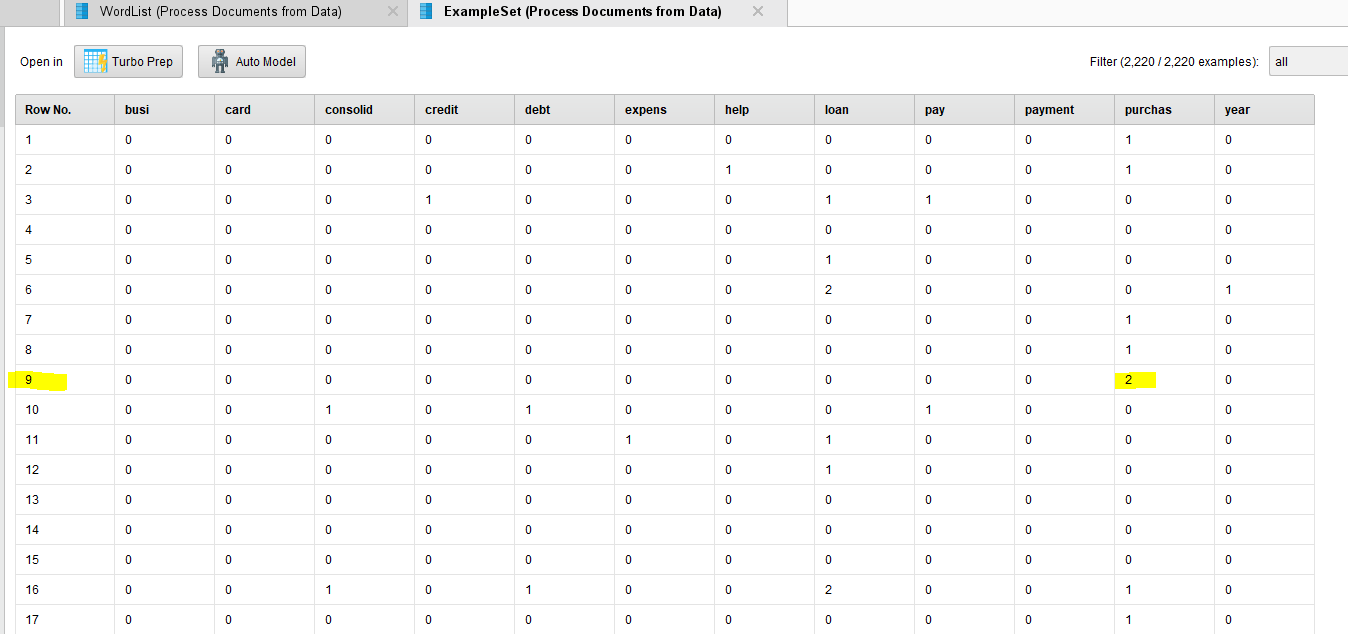
En example set vemos el resultado de la búsqueda binaria de cada palabra por documento (registro), por ejemplo, en el registro 10 vemos que se encuentran tres palabras consolidar-deuda-pagar, la cual a simple vista podríamos discernir el motivo de este préstamo el cual es una consolidación de deudas.



Para el método TF-IDF se muestra la relevancia de cada palabra dentro de cada documento, y para el mismo caso del documento 10 vemos que la palabra con menos peso es pagar, para la cual si extrajéramos esta nos quedaría consolidar-deuda que nos seguiría brindando el mismo significado.



En el caso de Term Occurrences, pienso que no nos brinda mayor significancia que TF-IDF ya que nos da un resumen de conteo de palabras por documento, aunque si estableceríamos la importancia de una palabra dentro de un documento basado en su número de apariciones los resultados serian muy distintos a los encontrados en TF-IDF, por ejemplo el documento 10, en este caso las ocurrencias son iguales para las tres palabras por lo cual el peso de cada una seria de 0,33



¿Crees que binary term ocurrences es la mejor forma de representar los datos?

No, binary term occurrences nos brinda un análisis muy básico en el que únicamente podemos ver si la palabra se encuentra en un documento o no.

¿Cuál es la diferencia entre las dos vistas de los resultados (Example Set y Word List)?

WordList muestra un resumen del conteo por palabra, mientras que ExampleSet muestra un detalle de conteo por documento.

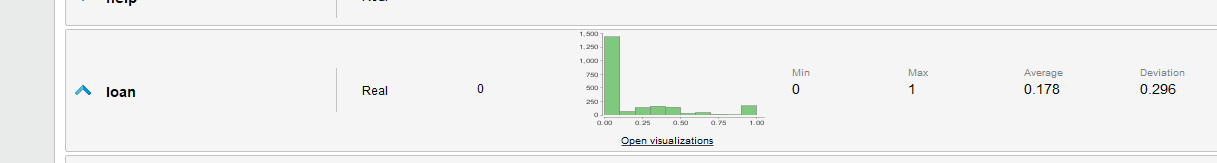
¿Cuál es la diferencia entre Total Occurences y Document Occurences?

Total occurrences es la cantidad de veces que una palabra se encuentra en el fichero, y document occurrences es el numero de veces que la palabra se encuentra en el fichero agrupada por documentos, es decir, si la palabra PAY se encuentra dos veces en un documento, suma 1 al document occurrences para PAY.

¿Cuáles serían los valores óptimos de podado (pruning) para quedarnos con palabras representativas?

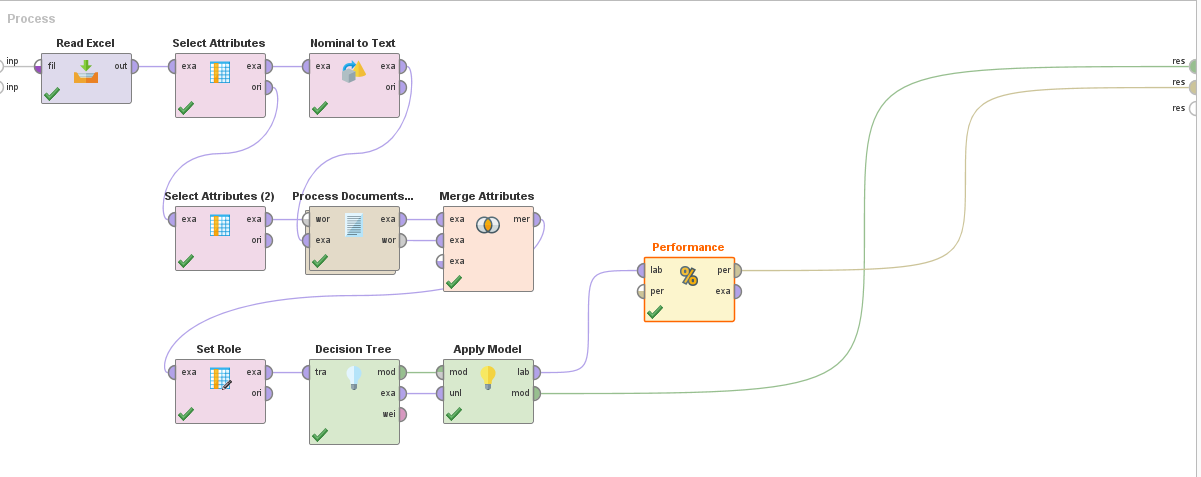
Considero que podar las palabras menores que 3 caracteres es un buen acercamiento, ya que por ejemplo hay palabras de 3 caracteres que, si pueden llegar a brindar mayor relevancia a alguna palabra dentro del documento, por ejemplo “aun” podría darle mayor relevancia a una palabra predecesora, y “por” a una palabra sucesora.

En la vista Example Set, elige algún término representativo y abriendo la pestaña statistics y luego visualization, muestra el histograma de ese término ¿Qué representa?



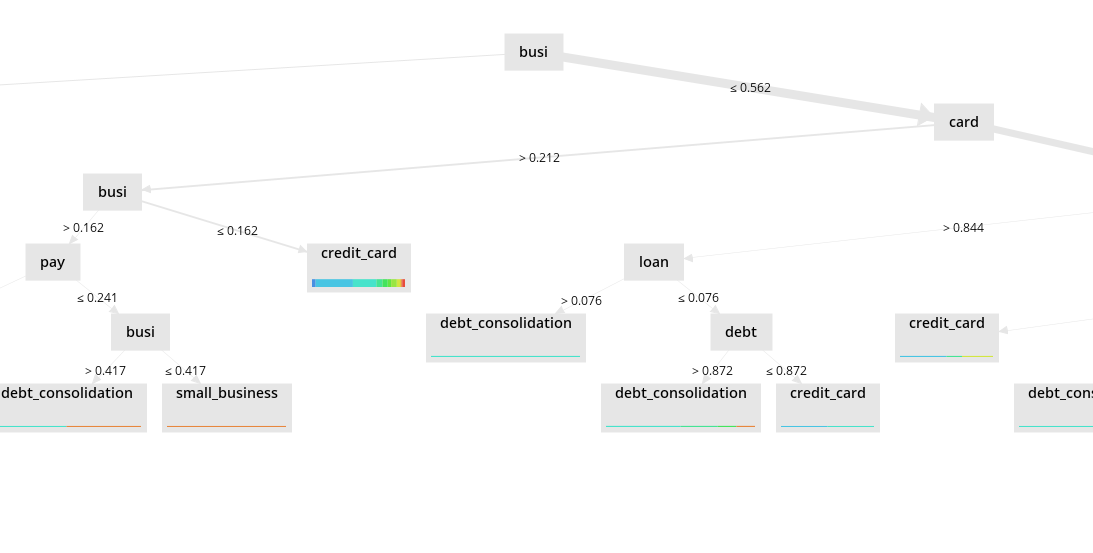
En el caso de TF-IDF muestra un resumen de relevancia por palabra, mostrando su mínima y máxima relevancia junto con la media de relevancia con la cual podríamos determinar la palabra más relevante en el fichero.

Entrega una impresión de pantalla del proceso configurado.



¿Qué resultados de clasificación obtienes?

Como resultados tenemos una matriz y un árbol donde se clasifican las relaciones entre las palabras en la descripción del préstamo, según el árbol, la palabra padre con la que se llega a una clasificación optima para este caso es busi, siendo sus hijos inmediatos card y payment. Adicional obtenemos el un acierto en la predicción del modelo que llega al 33.24% considerado como bajo ya que de cada 3 hojas acierta la rama padre de una sola.



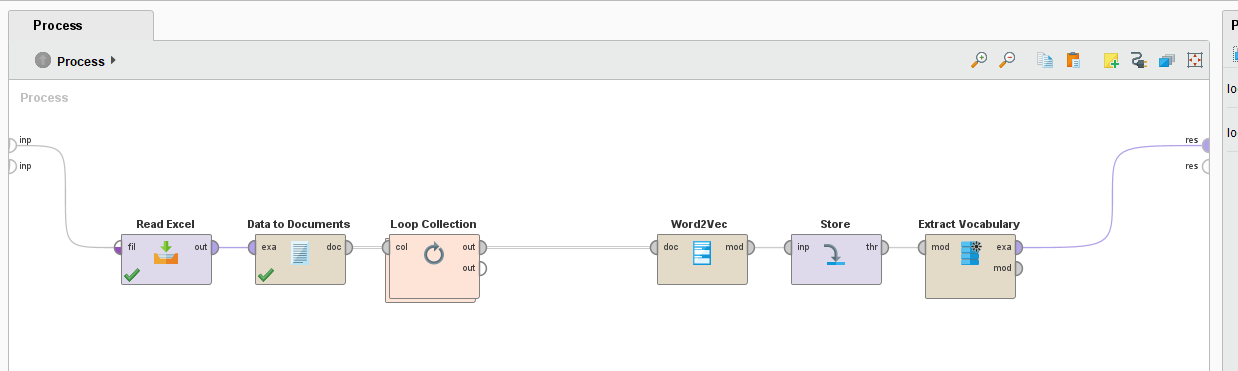
Prueba diferentes representaciones de los documentos (configuración del operador process text) e indica con cuál de ellas se obtienen los mejores resultados ¿Por qué crees que es así?

**Generación de la representación con Word embeddings**

Al correr el proceso sale un error que pide tokenizar los documentos, pero dentro del loop collection estoy aplicando el operador de tokenize y he probado lor diferentes para metros de dicho operador:



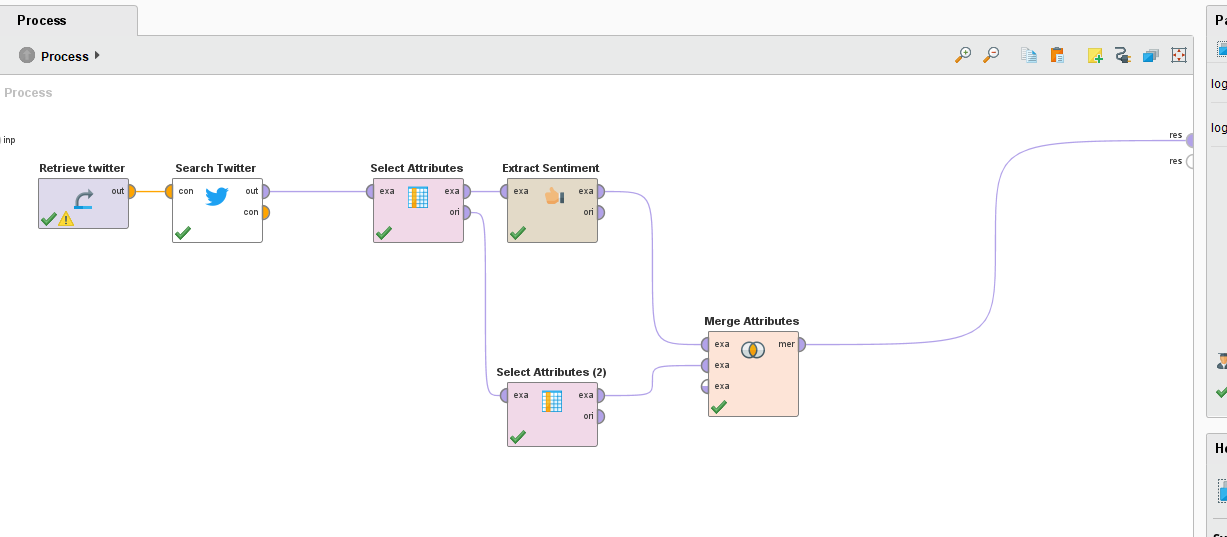
Finalmente, el proceso se visualiza de la siguiente forma:



Según entiendo con este proceso el objetivo es generar un corpus a través de la entrada de motivos de préstamo, para que posteriormente se puedan generar cálculos en sentencias procedentes de otros motivos de préstamo.

**Obteniendo datos de twitter**

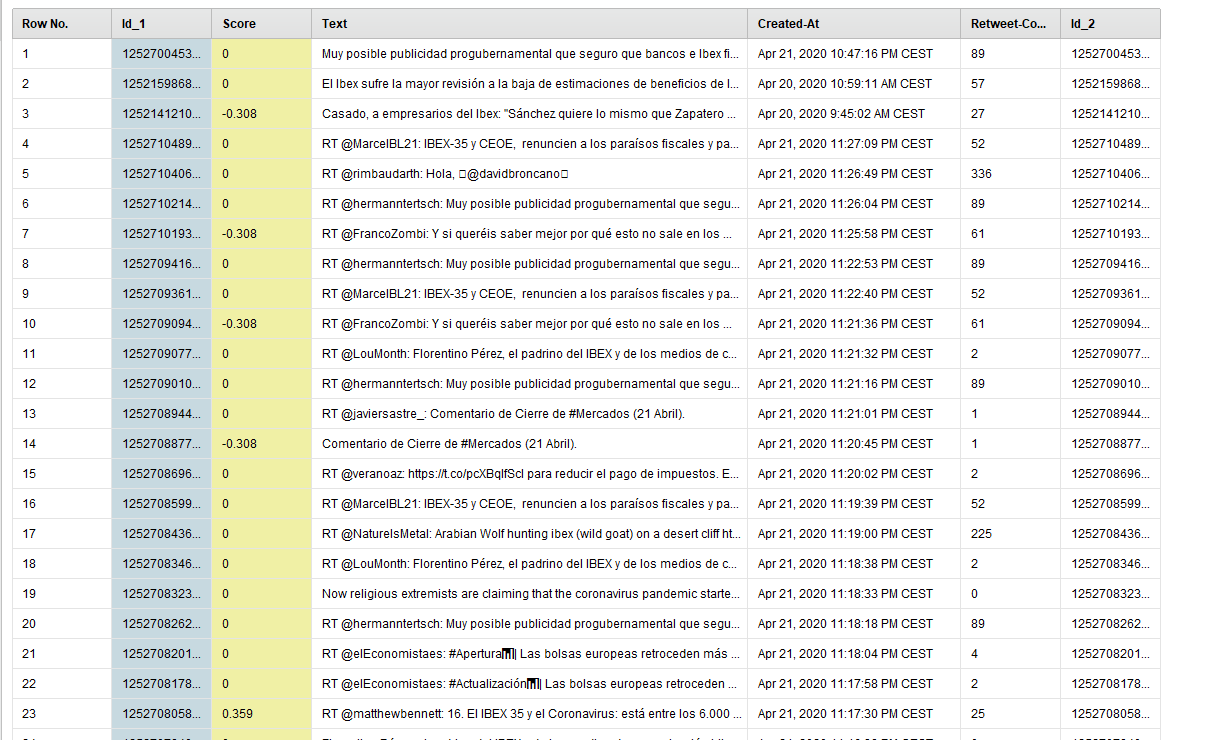
Se busca la palabra IBEX dentro de los Tweets con el objetivo de hacer un análisis de sentimiento sobre la cotizada quedando el siguiente modelo:



Se extrae el score del sentimiento configurado de la siguiente manera:

Muy positivo = 1, Positivo = 0.75, Neutral = 0, Negativo = -0.75 and Muy negativo = -1

Adicional se selecciona el attributo de conteo de retweets para posteriormente analizar la importancia y pos tanto un posible impacto de la subida o bajada del precio del índice. Se obtienen los siguientes resultados:



Como se aprecia la mayor cantidad de resultados son de tipo neutral por lo que considero se podrían podar para quedarnos con los positivos y negativos.

El modelo aquí presente se complementaria con un proceso de extracción de data financiera, ya sea de Google finance o yahoo finance para obtener el precio de cierre y calcular su volatilidad para que fuese comparada con una serie temporal de score de sentimientos y así poder verificar el impacto que tienen los tweets relevantes en el precio de una cotizada.