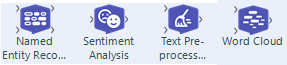
**TEXT MINING**

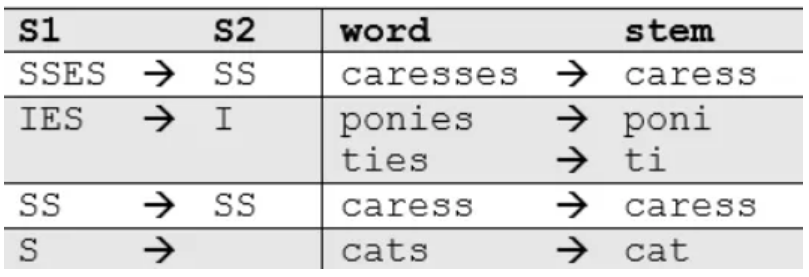
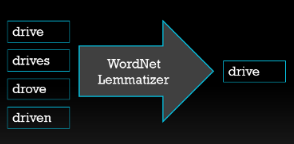
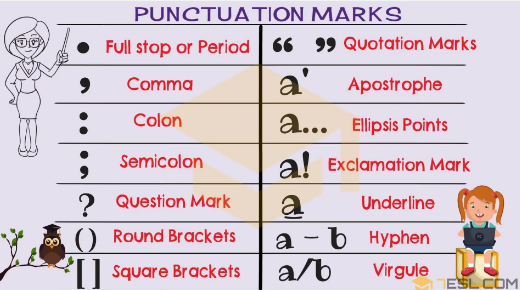
**Created By: Pushkarni S Rao**

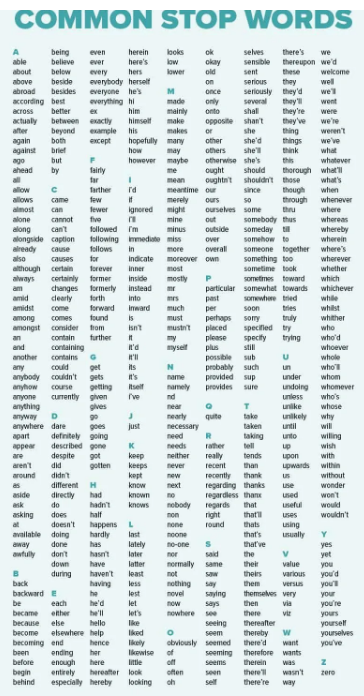
**Text Mining**

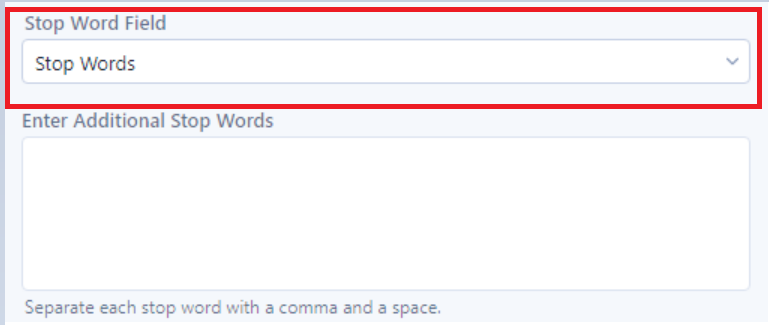
* Text mining in Alteryx is a powerful tool for extracting insights from unstructured text data.
* It can be used for sentiment analysis, categorizing text data into predefined topics, extracting important keywords, and preprocessing text data for analysis.
* Alteryx provides data visualization capabilities for visualizing the results of text mining, making it easier to explore patterns and trends.
* Text mining in Alteryx is beneficial for tasks such as customer feedback analysis, market research, brand monitoring, and content analysis.  
    
   **In this document I have explained these 4 tools under Text Mining:**

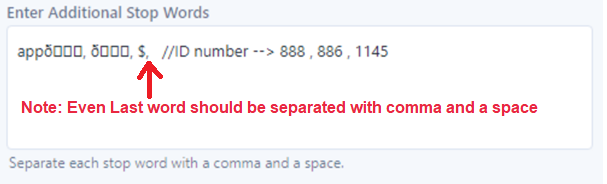


1. **Text Pre-Processing:**  
   **Use Text Pre-processing to clean up text data.**

* **Language:** Choose one language for analysis.   
  (English, Spanish, Portuguese, German, Spanish, French and Italian)
* **Text Field:** Choose which column will be processed.  
    
  **Note: If we don’t tick any of the below options the output will be separated by space when there is single punctuation between words.**
* **Text Normalization**  - **Convert to word root**: modifies word to base form.  
   **Lemmatize**---> Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item.  
  Example: “Tested”, “testing” becomes “test”.   
    
    
    
    
  
* **Filter** - **Digits:** removes numbers.  
   - **Punctuation:** removes everything which is mentioned below  
  

- **Stop Words:** removes words like “the”,”a”,”is”,”are” etc.  
**Note: We need to toggle Use Default stop words (Only then common stop words will be removed from the sentence)**  


* **Stop Word Field**: Select the field in the drop down.  
    
  **Note: Stop Word Field drop down section will appear on the configuration window only if we pass any data through the S input.**
* **Additional stop words**: custom listing of words to remove.  
  We can add any additional words here which we want to eliminate manually other than common stop words.  
    
  **Note: We can eliminate words passing through the S input and eliminate using the Enter Additional Stop words section. Both does the same job.**

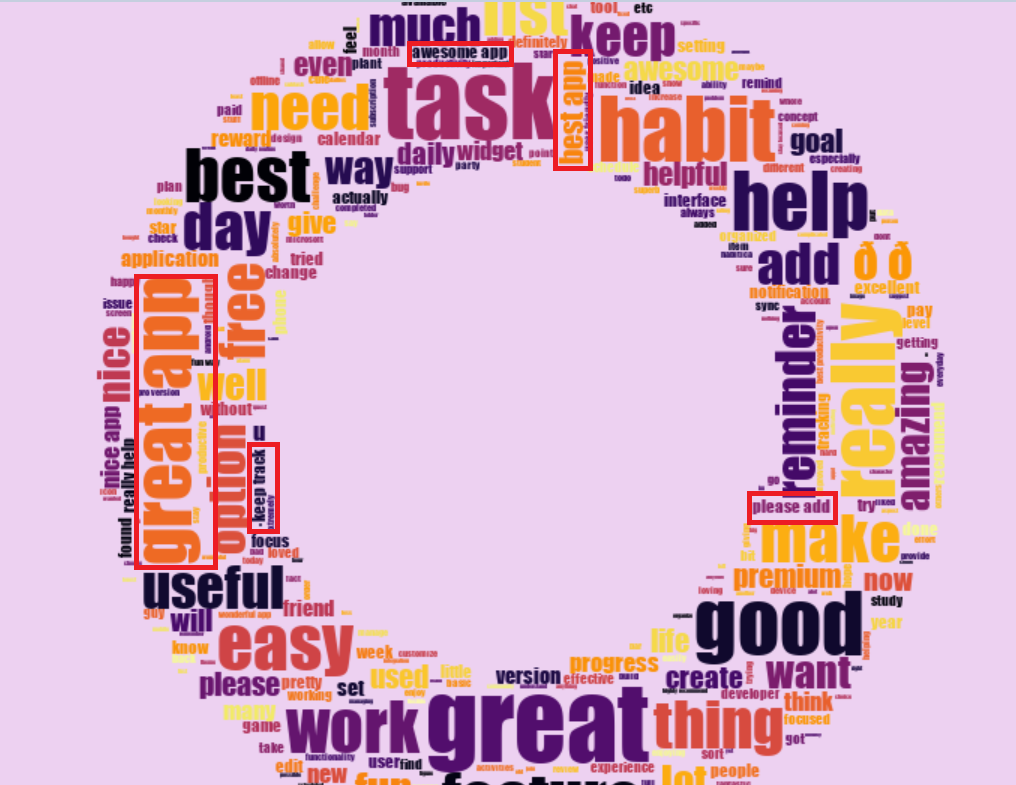


**Note: Even Last word should be separated with comma and a space**

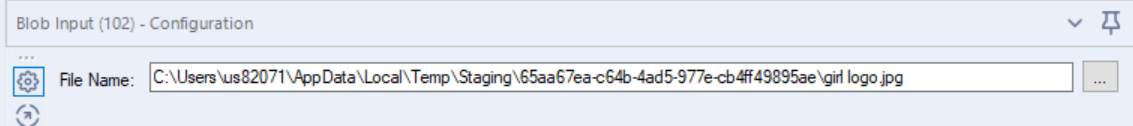
**2) Sentiment Analysis:**  
  
Sentiment analysis tools allow for the examination of text values to determine whether the writer’s sentiment was positive, neutral, or negative. This function group includes tools to pre-process text, others to display key words in word clouds or in topic reports, **(Objectively evaluates emotions in text)**

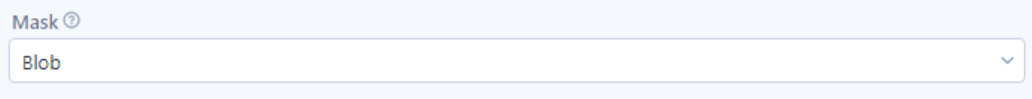
* **Language:** Choose one language (English)
* **Algorithm:** VADER  
    
  Sentiment Analysis is based on VADER Algorithm  
  (VADER = Valence(va·luhns) Aware Dictionary for Sentiment Reasoning)  
   - The valence of emotion refers to whether it is positive or negative.  
   - The magnitude of emotion refers to how positive or negative it is.  
   - VADER also identifies text that is not emotional or neutral in its valence.  
     
  **Note: Don’t use text pre-processing tool before sentiment analysis tool as it will remove punctuation.**
* **Text field:** Choose the column to examine
* **Additional Columns:  
   - Find Sentiment at Sentence Level -->**  This will give better results as it will break down the data at commas or delimiters.  **- Output Categorical Sentiment -->** This will say if the sentence is positive, negative, or neutral based on **Compound Sentiment Score.**
* **If we select Compound Sentiment Score, we get 2 options to set:  
  Max/Min Classification:   
   -** We can set the value between -1 to 0 for negative.  
    **-** We can set the value between 0 to 1 for positive.  
    **-** The range of values between negative and positive that will be considered as neutral.
* **Result window:**  
  **Compound Sentiment Score:   
   -** Value <0 negative  
    **-** Value >0 positive  
    **-** Value =0 neutral  
  **Final Result:**You can then use the output of the Text Mining Tool in further analysis, such as visualizations, reporting, or predictive modeling.  
  **Note:   
  -Sarcastic sentences will be considered as positive not comprehensive, so we need to cross check, if the results are as expected.  
  - Emojis and slang are also supported.**

**3)** **Word Cloud:  
  
Use Word Cloud to visualize text data.**

* **Field:** Only required field for word cloud.
* **Customize Word Cloud:** If selected optional values are available.
* **Background color:** Background color for word cloud color.
* **Theme:** Set color schemes for the word cloud values.
* **Generate Phrases:** Here the tool will treat two-word phrases as single phrase. **(A phrase is a short group of words conveying a concept and is usually part of a longer sentence.)** 
* **Max words:** The maximum number of words to be displayed in Cloud.  
  Repeated words will be displayed. If certain words are repeated many times, it will be displayed in larger font and less often are going to be smaller words around the edges.
* **Size of word Cloud:**  
   - Width in px  
   - Height in px
* **Mask:** A mask is an image used to define the shape of the word cloud.   
  - You must input the mask into the "M" anchor of the Word Cloud tool.  
  - Mask must be imported as a BLOB.   
  - You can use the Blob Input tool(Developer Palette) to import the mask as a BLOB.

**Steps to add image:  
  
-** To take an image as a template, add a Blob Input tool (binary large object) from the Developer tool category and select the path where the file is located.  


  
**-** Once this is done, in the Word Cloud configuration, the Blob option will appear in the mask option.

  
  
**-** Add the browse tool to view the visual. (I had considered Alteryx logo)  
  
  
  
  
**Note: Visual representation of text is based on size of the raw image**.

**4) Named Entity Recognition:  
  
It identifies entities like people, places, and things in text.**

* **Language:** Choose one language for analysis.   
  (English, Spanish, Portuguese, German, Spanish, French and Italian)
* **Column with Text:** Choose which column will be processed.  
    
  **Optional Filters**
* **Train with New Entities (Select the Box)  
    
  Note: E input anchor (this is optional input anchor, but if we want to train with new entities, provide them in data connected to the E anchor): Connect the data with the custom entities you want to identify. This data must contain the custom entity names and labels you want to use to train the model.**  
  **Match Entities:**
* **Column with Entities:** From the custom entity list connected to the E input anchor.  
    
  **Default English Entity List**

PERSON: Fictional and non-fictional people.

NORP: Nationality, religion, or political group.

FAC: Facilities such as buildings, airports, highways, and bridges.

ORG: Organizations such as companies, agencies, and institutions.

GPE: Geographical entities such as countries, cities, and states.

LOC: Non-GPE locations such as mountain ranges, bodies of water, and continents.

PRODUCT: Products such as vehicles and foods. Excludes services.

EVENT: Events such as named hurricanes, wars, and sports events.

WORK\_OF\_ART: Works of art such as books, songs, and movies.

LAW: Named documents made into laws.

LANGUAGE: Named languages.

DATE: Date entity.

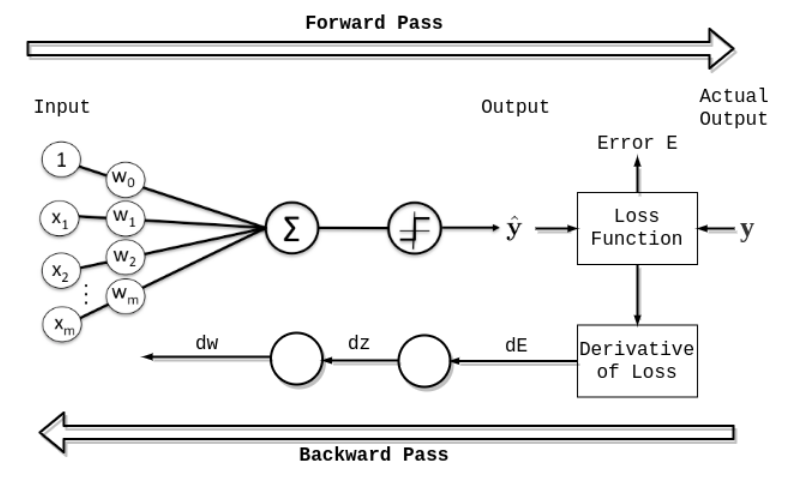
TIME: Time entity, less than a day.

PERCENT: Percentage, includes "%" and the word "percent."

MONEY: Monetary value, includes the unit.

QUANTITY: Measurements such as height, weight, and distance.

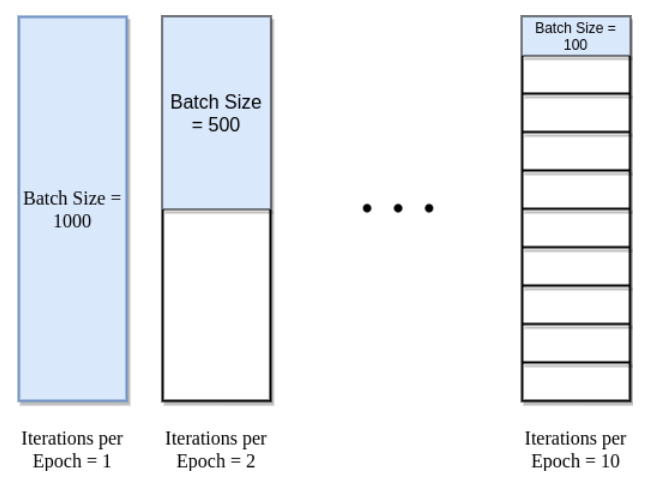
ORDINAL (**aw**·duhn·uhl): Ordinal entities such as first, second, and third

* **Column with Labels:** From the custom entity list connected to the E input anchor.
* **Case Sensitive (Select the box if you want your model to be Case Sensitive)  
    
  Train Model:**
* **Epochs (ee-pawk) and Batch size:**  
  **An epoch means training the neural network** (Neural networks are used to mimic the basic functioning of the human brain and are inspired by how the human brain interprets information. It is used to solve various real-time tasks because of its ability to perform computations quickly and its fast responses.) **with all the training data for one cycle. In an epoch, we use all the data exactly once. A forward pass and a backward pass together are counted as one pass:**  
    
  

**Working of this diagram:**

We feed the data through the inputs, the data goes through a forward pass and then we obtain an output. With the output we can calculate a loss. Then we will use backpropagation to attribute some fault to each model parameter for the resulting error (loss). Then we will use gradient descent algorithm to update the model parameters accordingly.

**Note: Since one epoch is too big to feed to the computer at once we divide it in several smaller batches.**So in terms of Epochs and Batch, one epoch ends when all the training data available has been consumed. The second epoch goes through all the data again. In a simple neural network with not much data, you will pass all the training instances through the network successively and get the loss for each output. Then we will get an average of these losses to estimate the total loss for all instances. This results in one backpropagation per epoch.

However, most of the time it is not possible to fit all the data into memory so we must use batches, this means we will only feed-forward some training instances at a time. Then we will calculate the loss resulting from these instances and tune the parameters using backpropagation.   
  
Say we have 1000 data points as presented in the figure below:  
  
  
 **Note: Iteration: Iterations is the number of batches needed to complete one epoch.**  
**For example:** Let’s have the training dataset having 1000 training samples. And we want to break the dataset into a batch size of 100. Suppose we are going for 5 epochs, Then the total number of iterations will be :

Total number of training samples = 1000

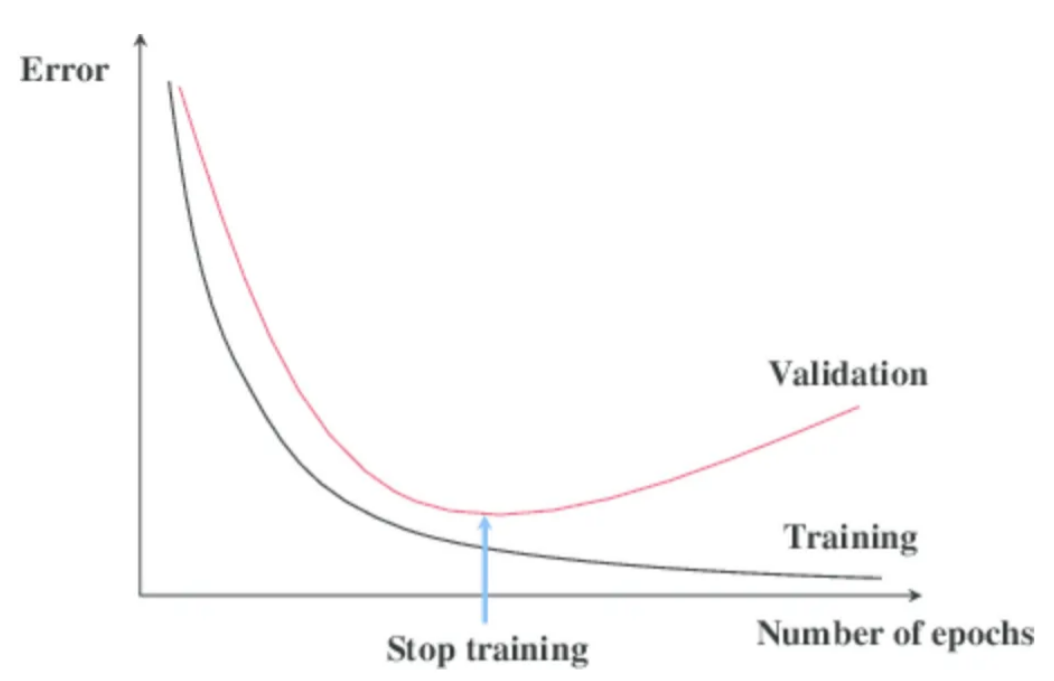
Batch size = 100

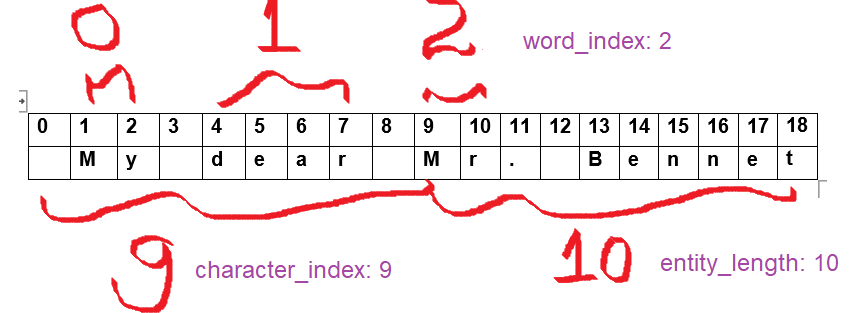
Total number of iterations=Total number of training samples/Batch size=1000/100=10

Total number of iterations = 10

One epoch = 10 iterations

Total number of iterations in 5 epochs = 10\*5 = 50 iterations

* **Batch Size:   
    
  Total number of training samples present in a single batch.**We cannot pass the entire dataset into the neural network at once. So, you divide dataset into Number of Batches or sets or parts.
* **Early Stopping: (Select the box if you want to early stop)  
    
  Early stopping is a strategy for avoiding “overtraining” your model**.   
    
  - We divide our data into two sets for training machine learning models:  
   the training set and the validation (or test) set.   
  - The first is employed for training, while the latter is used to evaluate how effectively the model is functioning.  
  - Simply, if the model stops developing and begins to perform poorly during training, we cease training. As a result, we “early end” the model.- Early Stopping is a very different way to regularize the machine learning model. The way it does is to stop training as soon as the validation error reaches a minimum.   
    
  **The figure below shows a model being trained:**  
    
    
  As the epochs go by, the algorithm leans and its error on the training set naturally goes down, and so does its error on the validation set. However, after a while, the validation error stops decreasing and starts to go back up. This indicates that the model has started to overfit the training data. With Early Stopping, you just stop training as soon as the validation error reaches the minimum.
* **Output:**

1. The D output anchor adds 2 columns to the output:
2. **entities:** This column contains a JSON output with a list of entity tags and descriptions.
   * + **entity:** Entity found by the model.
     + **label:** The entity label.
     + **character\_index:** The index of the 1st character of the word in the body of text. The index starts at 0.
     + **word\_index:** The index of the word in the body of text. The index starts at 0.
     + **entity\_length:** Character length of the entity.  
         
       **Example 1:** Chapter\_1  
        {"entities": [{"entity": "Chapter 1", "label": "LAW", "character\_index": 0, "word\_index": 0, "entity\_length": 9}]}  
         
       **Example 2:** “ My dear Mr.\_Bennet , ” said his lady to him one\_day , “ have you heard that Netherfield\_Park is let at last ? ”  
       {"entities": [{"entity": "Mr. Bennet", "label": "CHARACTER", "character\_index": 9, "word\_index": 2, "entity\_length": 10}  
         
       
3. **entity\_diagram:** This column contains your text with labeled entities and is visible with the Browse tool.
4. The M output anchor contains a model object.  
   **Note: You can save the model object and use it on new data with the Predict tool.**