# Article: 7 Steps of Machine Learning- (<https://livecodestream.dev/post/2020-06-02-7-steps-of-machine-learning/>)

* Creation of diverse data sets for our features of interest
* The step of gathering data is the foundation of the machine learning process
* Mistakes such as choosing the incorrect features or focusing on limited types of entries for the data set may render the model completely ineffective
* Another major component of data preparation is breaking down the data sets into 2 parts. The larger part (~80%) would be used for training the model while the smaller part (~20%) is used for evaluation purposes.
* This is important because using the same data sets for both training and evaluation would not give a fair assessment of the model’s performance in real world scenarios.
* Apart from the data split, additional steps are taken to refine the data sets. This could include removing duplicate entries, discarding incorrect readings etc.
* supervised learning models-In such models the outcome is known so we continuously refine the model itself until our output reaches the desired accuracy level.
* unsupervised learning -  the outcome is unknown and we need classification to be done
* reinforcement learning - learning to make better decisions on the basis of trial and error
* Training requires patience and experimentation
* With the model trained, it needs to be tested to see if it would operate well in real world situations. That is why the part of the data set created for evaluation is used to check the model’s proficiency.
* If the results are not satisfactory then the prior steps need to be revisited so that root cause behind the model’s underperformance can be identified and, subsequently, rectified.
* If the evaluation is not done properly then the model may not excel at fulfilling its desired commercial purpose.
* Hyperparameter tuning - improve upon the positive results achieved during the evaluation step
* Prediction -  the stage where we consider the model to be ready for practical applications.

# Arcticle: How to create text classifiers with Machine Learning(<https://monkeylearn.com/blog/how-to-create-text-classifiers-machine-learning/>)

# You need to define the tags that you will use, gather data for training the classifier, tag your samples, among other things.

# What are the tags that you want to assign to your texts? This is the **first question** you need to answer when you start working on your text classifier.

# Sometimes you know which are the tags you want to work with (for example if interested in sentiment analysis), but sometimes you don't know what tags you should use. In these cases, you need to first explore and **understand your data** to determine what are appropriate tags for your model.

# When you want to be more specific and use subtags, you will need to define a **hierarchical tree** that organizes your tags and subtags.

# A crucial part of this process is giving a proper structure and criteria to your tags.

# each set of subtags needs to be implemented on a separate classifier.

# Tips for defining your tags

1. **Avoid overlapping**

* Use disjoint tags and avoid defining tags that are ambiguous or overlapping
* **there should be no doubt in which tag a text should be placed**
* Overlapping between your tags will confuse to your model and affect the accuracy of the predictions negatively.

1. **Don't mix classification criteria**

* Use **one single classification criteria per model**.

Eg:- one to classify a company according to who are their customers (B2C, B2B, Enterprise)

another model to classify a company according to the industry vertical it operates (Finance, Media, Construction).

1. **Structure**

* Organize your tags according to their **semantic relations**.
* A classification process that has a clear structure can make a significant difference and will be a huge help to make accurate predictions with your classifiers.

1. **Start small and then go big**

* If it's your first time training a text classifier, we recommend starting with a simple model.
* **Start with a small number of tags (<10).**
* When you get this simple model to work as expected, try adding a few more tags and work in your model until the new tags are accurate enough. Eventually, you can keep iterating adding more tags as you need.
* You can use open data from sites like [Kaggle](https://www.kaggle.com/datasets), [Quandl,](https://www.quandl.com/" \t "_blank) and [Data.gov](https://www.data.gov/).
* [Sentiment analysis is a much harder problem](https://monkeylearn.com/blog/sentiment-analysis-apis-benchmark/) to solve and it needs much more text data.
* **We suggest starting by tagging at least 20 samples per tag and take it from there.**
* For topic detection, we have seen some accurate models with 200-500 training samples in total.
* Sentiment analysis models usually need at least 3,000 training samples to start to start seeing an acceptable accuracy.
* It's much better to start with fewer samples, but being 100% sure that those samples are representative of each of your tags and are correctly tagged than to add tons of data but with lots of errors.
* Some of our users add thousands of training samples at once (when are creating a custom classifier for the first time) **thinking that the high volumes of data is great for the machine learning algorithm, but by doing that, they don't pay attention to the data they use as training samples**. And **most of the times many of those samples are incorrectly tagged.**
* better to **start with few but high-quality training samples that are correctly tagged** and take it from there.
* **Accuracy - classifier distinguishes between its tags**.
* **Tips for improving accuracy-**

1. **Add more training samples** to its tags.
2. **Retag** samples that might be incorrectly tagged
3. **Merging the ambiguous tags**

* Accuracy on its own is not a good metric; you also have to take care of precision and recall. You can have a classifier with outstanding accuracy but still have tags with bad precision and recall.
* **low precision - false positives**, **other sibling tags were predicted as this** **tag**
* **low recall - samples from this tag were predicted as other sibling tags, false negatives**
* Usually, there's a trade-off between precision and recall in a particular tag, that means, if you try to increase precision, you could end up doing that at the cost of lowering recall, and vice versa.
* **Tips for improving precision and recall**

1. Explore the false positives and false negatives of your model.
2. If a sample was initially tagged as tag X but was correctly predicted as tag Y, **move that sample** to tag Y.
3. If the sample was incorrectly predicted as tag Y, try to make the classifier **learn more about that difference** by adding more samples both to tag X and tag Y.
4. Check that the keywords associated with tags X and Y are correct (see Keyword Cloud section to see how to fix that).