## ***Feature and Label Extraction (TF IDF / Count Vectorizer)***

Model *machine learning* tidak dapat bekerja pada *features* yang tekstual sehingga *features* tekstual tersebut perlu diproses menjadi *numerical features*. Cara yang biasanya digunakan untuk melatih model tekstual adalah dengan menghitung frekuensi kata (*words*) tersebut pada dokumen atau korpus. Terdapat 2 teknik yang dapat digunakan, yaitu *Count Vectorization* dan TF IDF *Vectorization*.

1. *Count Vectorization*

Pendekatan ini secara sederhana, menghitung frekuensi kemunculan kata pada semua korpus dan membuat *vocabulary* dari kata. Misalnya, membuat *vocabulary* kata seperti *angry*, *happy*, *relaxed*, dan *sad*.

1. TF-IDF *Vectorization*

TF-IDF juga dapat digunakan untuk menghitung frekuensi kata. Dengan TF-IDF, maka dapat di-*assign* jumlah frekuensi kemunculan kata yang mengidentifikasikan pentingnya kata tersebut pada dokumen atau korpus. Nilai TF-IDF dari sebuah kata dapat ditentukan menggunakan persamaan berikut:

Dimana adalah bobot kata dalam dokumen , adalah *term frequency*, atau jumlah di . Dan untuk nilai IDF adalah sebagai berikut:

Dimana IDF adalah *inverse document frequency* dan merepresentasikan kemunculan dalam . Nilai yang terbesar dari terjadi ketika hanya muncul dalam sebuah dokumen dan pengaruhnya sangat besar.

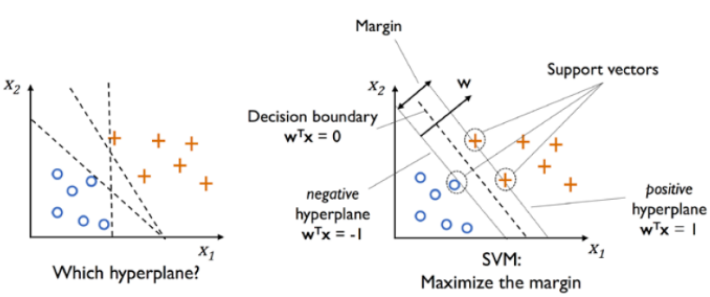
Selain itu, *feature* lainnya yang dapat digunakan untuk menentukan sentiment dari sebuah lagu berdasarkan liriknya adalah *Polarity Assignment of Lyrics*. Polaritas dari sebuah lirik lagu merupakan penjumlahan nilai polaritas dari semua kata yang terdapat dalam lirik. Kata-kata yang negatif biasanya memiliki nilai polaritas negatif dan kata-kata yang positif biasanya memiliki nilai polaritas positif. Dengan kata lain, bahwa lirik lagu dengan kategori *happy* akan memiliki jumlah kata-kata positif yang lebih banyak dibandingkan dengan kata-kata negatif. Dan sebaliknya, lirik lagu dengan kategori *sad* akan memiliki jumlah kata-kata negatif yang lebih banyak dibandingkan dengan kata-kata positif.

## ***Feature Selection***

Teks yang telah diekstrak selanjutnya perlu diproses dengan *feature selection* untuk mengetahui pola kata yang digunakan, urutan kata, dll. Urutan kata (*sequence of words*) merupakan elemen penting dalam melakukan analisis sentimen. Misalnya, kata “*i am sorry*” merupakan 3-gram yang artinya lebih masuk akal dibandingkan ketika analisis dilakukan per kata secara individu, misalnya “i”, “am”, “sorry”. Dalam pengerjaan proyek ini, akan dilakukan perhitungan pada *count vectorization*, TF-IDF *vectorization* dengan unigram, bigram, unigram & bigram untuk menghasilkan *document term matrix* dengan N Gram.

## ***Modelling with* SVM**

Setelah *feature-feature* yang akan digunakan diseleksi, selanjutnya adalah melakukan *train* pada dataset dengan menggunakan algoritma *Support Vector Machine* (SVM). SVM merupakan salah satu metode dalam *supervised learning* yang digunakan untuk klasifikasi (*Support Vector Classification*) dan regresi (*Support Vector Regression*). SVM bekerja dengan mencari *hyperplane* terbaik untuk memaksimalkan jarak antar kelas. *Hyperplane* adalah sebuah fungsi yang dapat digunakan sebagai pemisah antar kelas.



**Gambar 1. Support Vector Machine**

SVM dapat bekerja pada dataset yang memiliki dimensi tinggi menggunakan kernel trik sehingga tidak ada proses reduksi dimensi di dalamnnya. Berikut adalah beberapa keuntungan dari algoritma SVM:

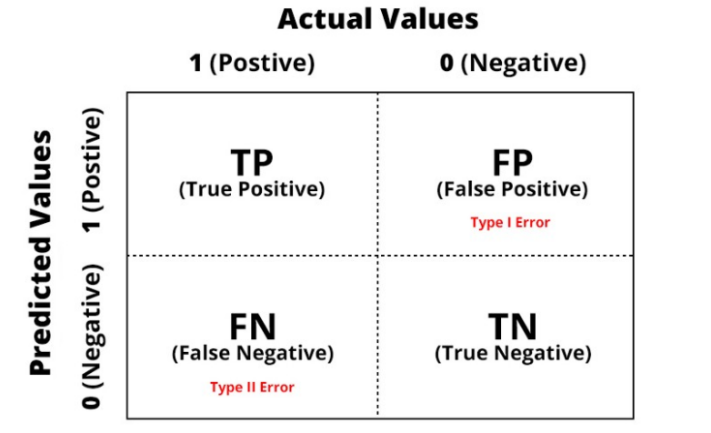
1. Efektif dalam ruang dimensi tinggi dan menggunakan *subset* dari *training points* pada fungsi keputusan yang disebut juga dengan *support vectors*, sehingga lebih hemat memori.
2. Efektif dalam kasus dimana jumlah dimensi lebih besar dari jumlah sampel
3. Dapat menggunakan fungsi kernel yang untuk pengambilan keputusan.

Sedangkan yang menjadi kerugian dari *Support Vector Machines* adalah sebagai berikut:

1. Algoritma SVM tidak secara langsung menyediakan estimasi probabilitas, sehingga perlu menggunakan *five-fold cross validation*.
2. Tidak bekerja dengan baik, ketika terdapat datanya besar karena memerlukan waktu yang lama untuk *training* seperti ketika dataset menjadi sangat *noise*, misalnya kelas target menjadi *overlapping*.

## ***Evaluation and Results***

Setelah model selesai dibangun, selanjutna adalah melakukan evaluasi. Pada tahapan ini proses evaluasi dari hasil yang didapatkan, dilakukan dengan *confusion matrix*. *Confusion matrix* memberikan informasi perbandingan hasil klasifikasi yang dilakukan model dengan klasifikasi yang sebenarnya.



**Gambar 2. Confusion Matrix**

Terdapat 4 istilah yang digunakan untuk merepresntasikan hasil klasifikasi, yaitu:

* 1. *True Positive* (TP) merupakan data positif yang diprediksi benar.
  2. *True Negative* (TN) merupakan data negatif yang diprediksi benar.
  3. *False Positive* (FP) merupakan data negatif namun diprediksi sebagai data positif.
  4. *False Negative* (FN) merupakan data positif namun diprediksi sebagai data negatif.

Untuk mengukur *performance metrics* dari *confusion matrix* dapat dilakukan dengan menghitung nilai *accuracy, precision, Recall,* dan F1-*Score*.

1. *Accuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar.
2. *Precision* menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan oleh model.
3. *Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi.