Math of Music Recommendation Systems

Liz Baranes, Emily Fernandez, Ananya Gandhi, Emilie Xu

THE RESEARCHERS

Liz
Baranes

Emily
Fernandez

Ananya
Gandhi

Emilie
Xu

PRIMARY REFERENCES

- Grosse, Roger. “Support Vector Machines and Boosting”. Machine Learning CSC 2515. Fall 2019. University of Toronto. https://www.cs.toronto.edu/~rgrosse/courses/csc2515_2019/readings/SVM-and-boosting.pdf
- Schapire, R.E. and Freund, Y.. “Boosting: Foundations and Algorithms,” 2012. The MIT Press. ISBN (electronic): 9780262301183. https://doi.org/10.7551/mitpress/8291.001.0001
- Shalizi, Cosma. “Logistic Regression”. Undergraduate Advanced Data Analysis 36-402. Spring 2012. Carnegie Mellon University. https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf
- Tian, H. Cai, H., Wen, J., Li, S. and Li, Y., "A Music Recommendation System Based on logistic regression and eXtreme Gradient Boosting," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-6, doi: 10.1109/IJCNN.2019.8852094. https://ieeexplore.ieee.org/abstract/document/8852094

TALK OUTLINE

- Background
  - Recommendation systems
  - Our approach to music recommendation
- Math
  - Logistic Regression (1944)
  - Support Vector Machines (1964)
  - Boosting (1995)
  - Lyrics Based Methods—LLMs
    - TF*IDF (1972)
    - Word2Vec (2013)
    - BERT (2018)
- Demo
- Results
- Future of the field

PROBLEM

Hard to find new songs + artists to listen to

RECOMMENDATION SYSTEMS: CURRENT STATE

- Recommendation Systems - data-driven algorithms that suggest additional products or services to customers
  - Based on purchase history, search history, demographic information, etc.
- Collaborative filtering - recommendations generated by analyzing the preferences of other users with similar interests/behavior

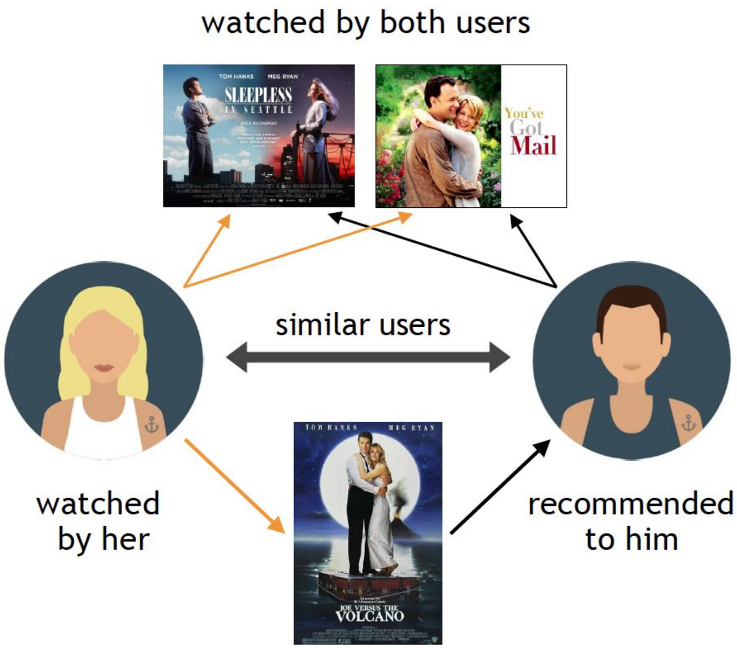


Image Credit: Recommendation Systems

RECOMMENDATION SYSTEMS: OUR APPROACH

- Content filtering - utilizes attributes and features of items to recommend other similar items to users
- Does not rely on data from multiple users to make recommendations
- Utilized content based filtering in our implementation

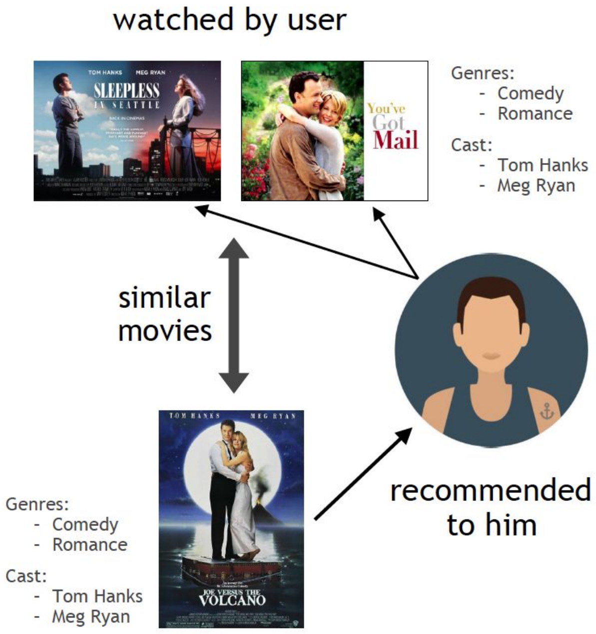


Image Credit: [Recommendation Systems]

ABOUT OUR DATASET

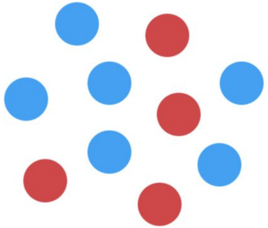
- Kaggle Most Streamed Spotify Songs 2023
    - 953 Rows with columns including:
        - Song title, artist name, release date, # of playlists, # of streams, key
    - Features (calculated through SVMs by Spotify) include:
        - bpm
        - danceability_%
        - valence_%
        - energy_%
        - acousticness_%
        - instrumentalness_%
        - liveness_%
        - speechiness_%
- Scrape lyrics from AZLyrics.com to create text-based features

LOGISTIC REGRESSION (1944)

GOAL: Predict the relationship between independent variable(s) and a categorical dependent variable → binary classification

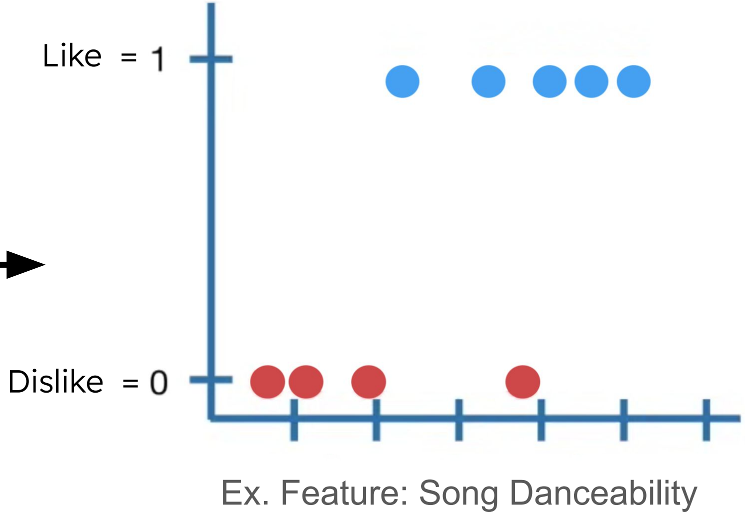
Image Credit: [Logistic Regression]

Labeled Training Data



Each dot is a song

= Like = Dislike

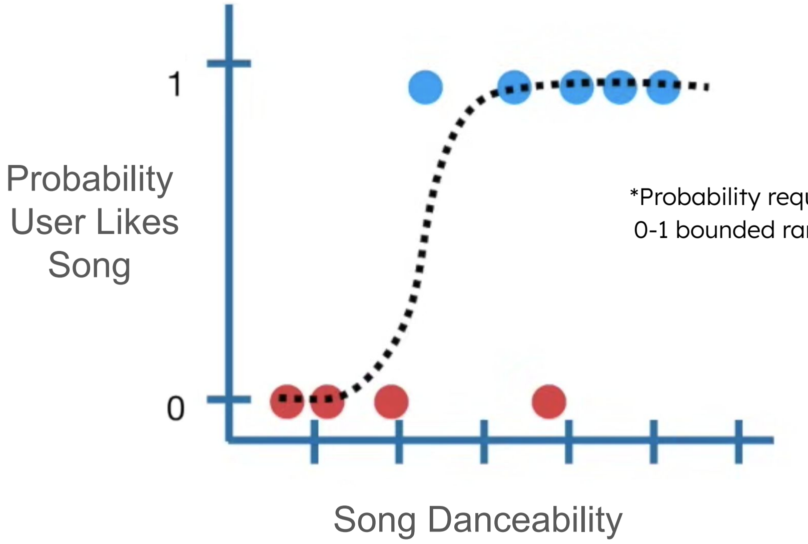


How do we model with a line?

LOGISTIC REGRESSION: S-CURVE

Fit data with a logistic function!

Example Logistic Function



*Probability requires 0-1 bounded range!

Sigmoid Function: $\mathbb{R} \rightarrow (0,1)$

"S-shaped"

$$p(x) = \frac{1}{1 + e^{-x}}$$

Models <u>conditional probability</u> that a user will like a song given a certain x feature

How do we find the best fit S-curve?

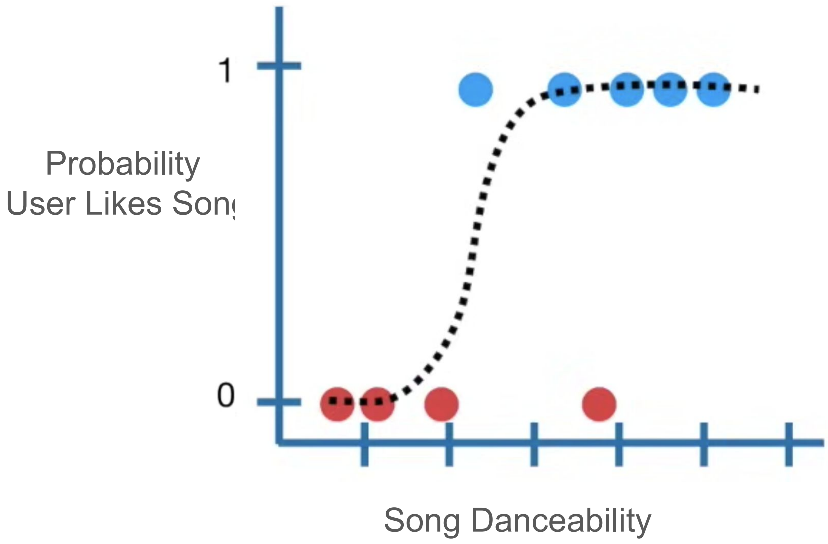
Image Credit: [Logistic Regression]

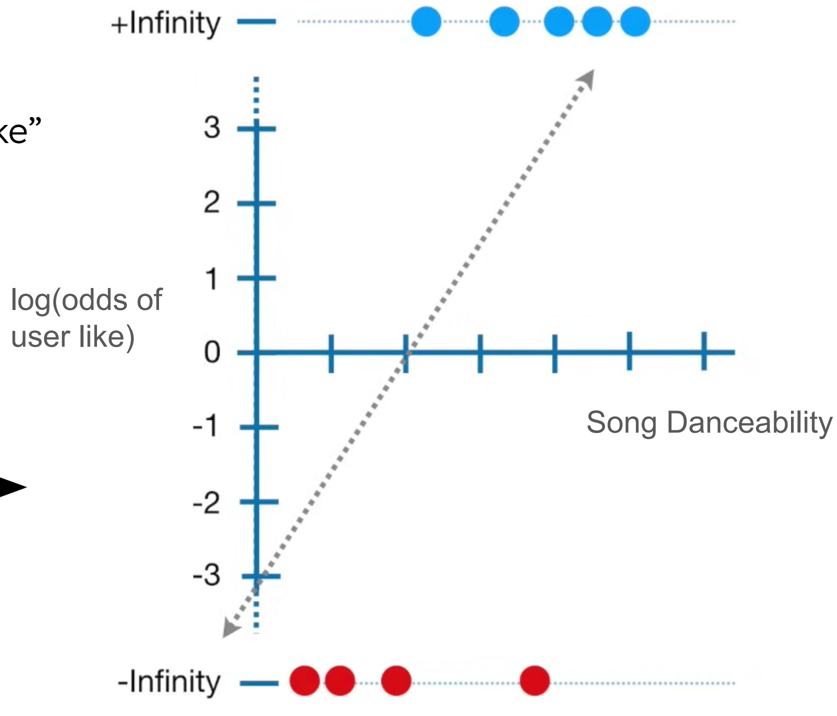
TRANSFORM S-CURVE

Take log odds of p

Y-axis “Probability of user like” to “Log odds of user like”

Induces linearity → data is no longer bounded (0,1)





$$\ln(\text{odds of event x}) = \ln\left(\frac{p(x)}{1-p(x)}\right)$$

Image Credit: Logistic Regression

LOGISTIC REGRESSION: LOG ODDS

Let vector x be a song with n features and $\mathrm{x} = (x_1,\ x_2, \ldots, x_n)$

Let conditional probability $p = P(Y = 1 |\ X = \mathrm{x})$ be the probability that the user likes song *x*

We will model this probability with a logistic function of the form

$$p = \frac{1}{1 + e^{-g(\mathrm{x})}}$$

where $g(\mathrm{x})$ is a function of song x and linear combination of $x_1, \ldots, x_n$

Let's explore the log odds of *p*

$$\mathrm{Odds}(p) = \frac{p}{1-p} = \frac{\frac{1}{1+e^{-g(\mathrm{x})}}}{1 - \frac{1}{1+e^{-g(\mathrm{x})}}} = \frac{1}{1 + e^{-g(\mathrm{x})} - 1} = \frac{1}{e^{-g(\mathrm{x})}} = e^{g(\mathrm{x})}$$

$$\ln(\mathrm{Odds}) = \ln\left(e^{g(\mathrm{x})}\right) = g(\mathrm{x})$$

Now we have a linear relationship. Let's visualize again

LOG ODDS: LINEAR CLASSIFIER

One feature model: $\operatorname{logit}(p) = \ln\left(\frac{p}{1-p}\right) = g(x)$

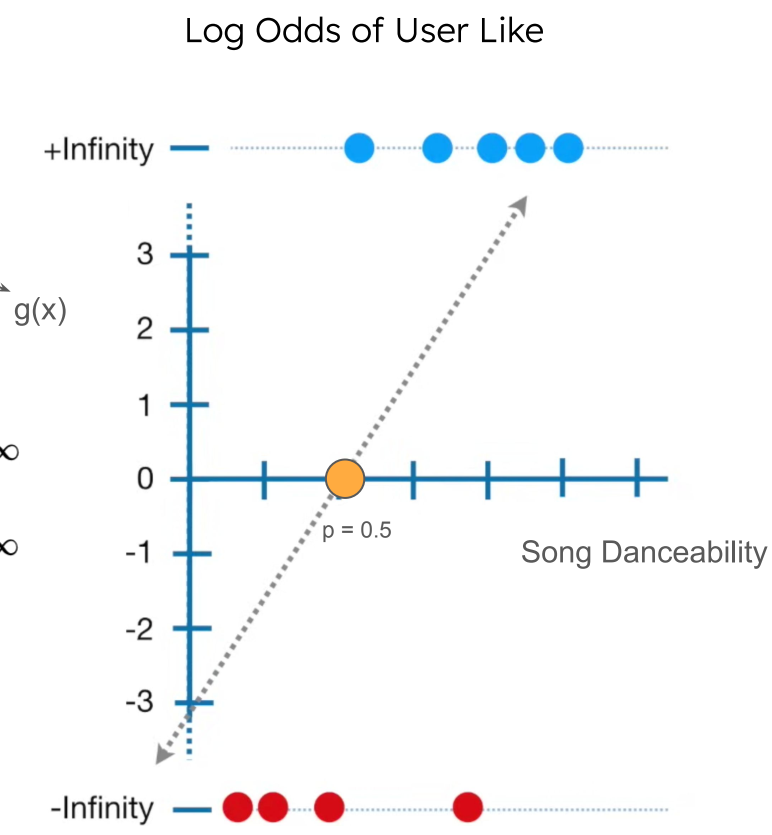
What happens to our boundary points?

$$\text{Max: } \operatorname{logit}(p=1) = \ln\left(\frac{1}{1-1}\right) = \ln\left(\frac{1}{0}\right) = \ln(1) - \ln(0) = +\infty$$

$$\text{Min: } \operatorname{logit}(p=0) = \ln\left(\frac{0}{1-0}\right) = \ln\left(\frac{0}{1}\right) = \ln(0) - \ln(1) = -\infty$$

$$\text{X-int: } \operatorname{logit}(p=0.5) = \ln\left(\frac{0.5}{0.5}\right) = \ln(1) = 0$$

Link function



Classification: g(x) ≥ 0 → like and g(x) < 0 → dislike

Image Credit: [Logistic Regression]

LOGISTIC REGRESSION: LINK FUNCTION

Vector x is a song with n features $\mathrm{x} = (x_1, x_2, \ldots, x_n)$ Probability that user likes song x: $p = P(Y = 1 | \mathrm{x})$

$$\mathrm{logit}(p) = \ln\left(\frac{p}{1-p}\right) = g(\mathrm{x}) \qquad \text{where } g(\mathrm{x}) = \mathrm{w}^{\mathrm{T}}\bar{\mathrm{x}}$$

$g(\mathrm{x})$ outputs log odds that user likes song x where feature x_i has weight w_i

$g(\mathrm{x}) = 0$ is the decision boundary separating the two classes

$$\ln\left(\frac{p}{1-p}\right) = w_0 + w_1 x_1 + \ldots + w_n x_n \longleftrightarrow p(\mathrm{x}) = \frac{1}{1 + e^{-\mathrm{w}^{\mathrm{T}}\bar{\mathrm{x}}}}$$

Now we must fit the line by estimating the weights. What is vector w?

MAXIMUM LIKELIHOOD ESTIMATION

Reminder of model:

$$\boxed{p(\mathrm{x}) = \frac{1}{1+e^{-\mathrm{w}^{\mathrm{T}}\bar{\mathrm{x}}}}}$$

Fitting the S-curve using MLE:

Likelihood of w = measure of "how probable are these weights given training data" → maximize this

Probability Mass Function (Bernoulli):

$$\left.\begin{array}{ll}\text{Prob. user like} & P(Y=1|\mathrm{x}) = p \\ \text{Prob. user dislike} & P(Y=0|\mathrm{x}) = 1-p\end{array}\right\} \; P(Y=y_i|\mathrm{x}) = p(\mathrm{x}_i)^{y_i}(1-p(\mathrm{x}_i))^{1-y_i}$$

Finding likelihood of weight w:

Likelihood function $L(\mathrm{w}) = \Pi_{i=1}^{m}\, p(\mathrm{x}_i)^{y_i}(1-p(\mathrm{x}_i))^{1-y_i}$ m = # training data points

Likelihood = product of probabilities because data points are independent

Now we have to maximize this function, though the product makes it more complicated…

Note: x_i feature is not to be confused with x_i sample

MAXIMUM LIKELIHOOD ESTIMATION

Reminder of model:

$$\boxed{p(\mathrm{x}) \;=\; \frac{1}{1+e^{-\mathrm{w}^{\mathrm{T}}\bar{\mathrm{x}}}}}$$

Likelihood $\quad L(\mathrm{w}) = \Pi_{i=1}^{m}\, p(\mathrm{x}_i)^{y_i}(1-p(\mathrm{x}_i))^{1-y_i}$

Apply the log-likelihood function (products → sums) $\quad l(\mathrm{w}) = \Sigma_{i=1}^{m} y_i \ln\left(p(\mathrm{x}_i)\right) + (1-y_i)\ln\left(1-p(\mathrm{x}_i)\right)$

Maximum of likelihood and log-likelihood will occur at the same point in the domain

We will maximize the log-likelihood by taking the partial derivative w.r.t. w and setting equal to 0

$$\begin{aligned}
\frac{\partial l(\mathrm{w})}{\partial \mathrm{w}} &= \frac{y}{p(\mathrm{x})}\frac{\partial p(\mathrm{x})}{\partial \mathrm{w}} + \frac{1-y}{1-p(\mathrm{x})}\cdot -\frac{\partial p(\mathrm{x})}{\partial \mathrm{w}} \\
&= \frac{\partial p(\mathrm{x})}{\partial \mathrm{w}}\left(\frac{y}{p(\mathrm{x})} - \frac{1-y}{1-p(\mathrm{x})}\right) \\
&= p(\mathrm{x})(1-p(\mathrm{x}))\cdot\bar{\mathrm{x}}\left(\frac{y}{p(\mathrm{x})} - \frac{1-y}{1-p(\mathrm{x})}\right) \\
&= (y(1-p(\mathrm{x})) - (1-y)p(\mathrm{x}))\cdot\bar{\mathrm{x}} \\
&= (y-p(\mathrm{x}))\cdot\bar{\mathrm{x}} \;=\; 0
\end{aligned}$$

$$\begin{aligned}
\frac{\partial p(\mathrm{x})}{\partial \mathrm{w}} &= \frac{\partial p(\mathrm{x})}{\partial(\mathrm{w}^{\mathrm{T}}\bar{\mathrm{x}})}\frac{\partial\left(\mathrm{w}^{\mathrm{T}}\bar{\mathrm{x}}\right)}{\partial \mathrm{w}} \\
&= \frac{\partial p(\mathrm{x})}{\partial(\mathrm{w}^{\mathrm{T}}\bar{\mathrm{x}})}\cdot\bar{\mathrm{x}} \\
&= p(\mathrm{x})(1-p(\mathrm{x}))\cdot\bar{\mathrm{x}}
\end{aligned}$$

*by derivative of sigmoid
$\sigma' = \sigma(1-\sigma)$ proof omitted

No closed form solution → proceed with numerical optimization method: gradient ascent, Newton's method, etc. to find max

SUPPORT VECTOR MACHINES (1964)

GOAL: Find some threshold in an N-dimensional space that distinctly classifies data points.

Hyperplane: Decision surface used to classify data points
Support vectors: Data points that lie close to the decision surface (hyperplane)

- Most difficult to classify
- Key to determining optimal location of hyperplane

Margin: Distance between support vectors

GOAL: Maximize the margin

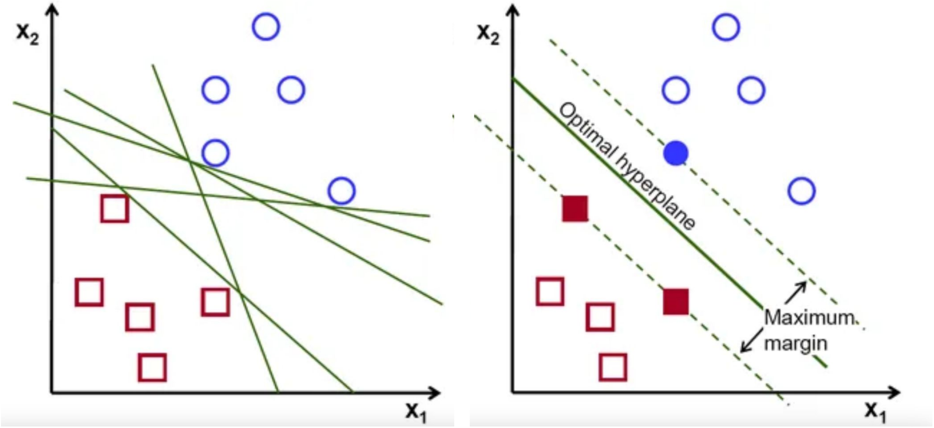


Image credit

SUPPORT VECTOR MACHINES

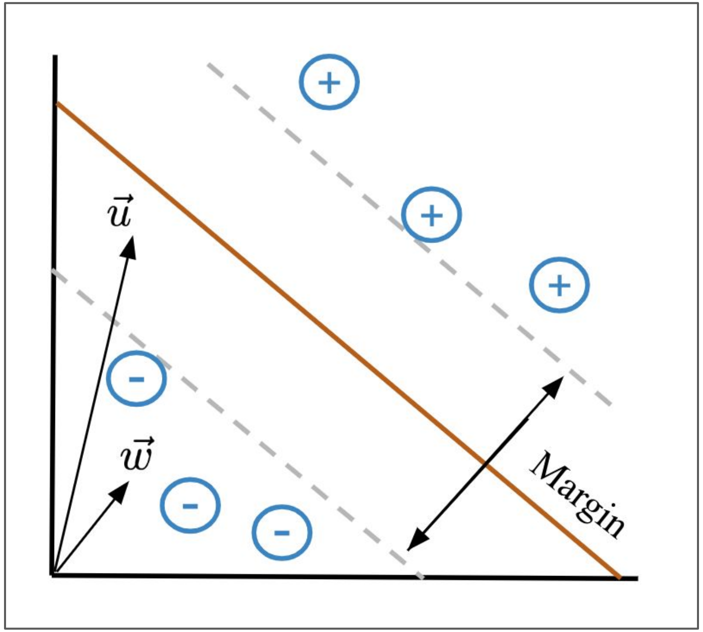
Take some vector $\vec{w}$ of any length perpendicular to the hyperplane, and some unknown vector $\vec{u}$ and consider $\vec{w} \cdot \vec{u} \geq C$

Decision Rule: If $\vec{w} \cdot \vec{u} + b \geq 0$ then $+$

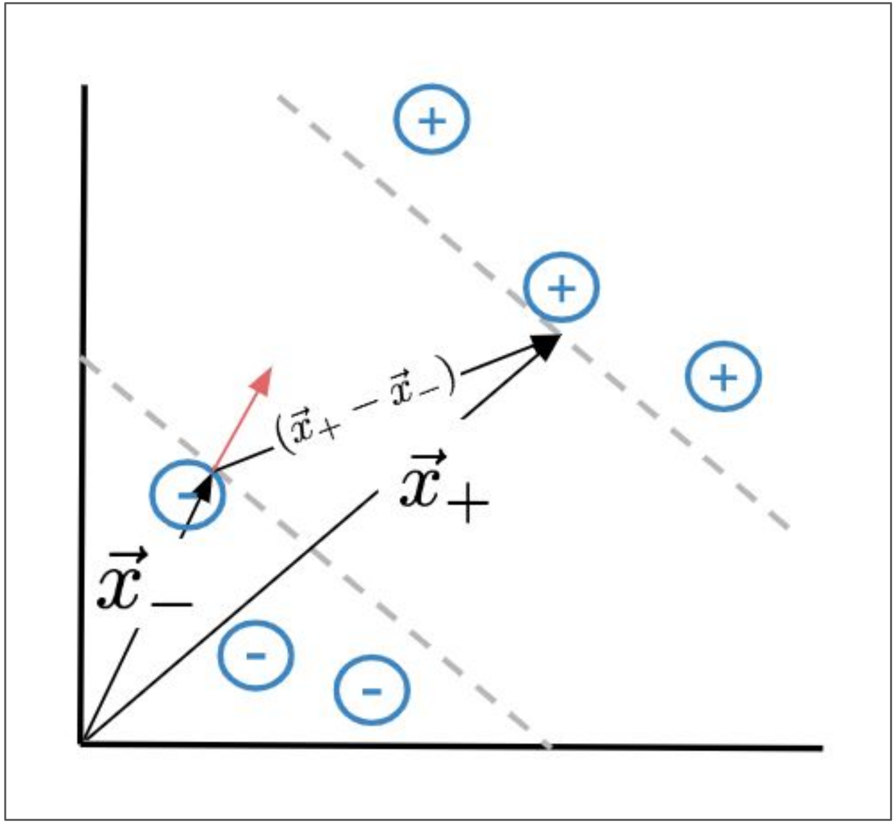
$$\vec{w} \cdot \vec{x}_+ + b \geq 1$$
$$\vec{w} \cdot \vec{x}_- + b \leq -1$$

$$y_i = +1 \text{ for positive samples}$$
$$y_i = -1 \text{ for negative samples}$$

$$y_i(\vec{x}_i \cdot \vec{w} + b) - 1 \geq 0$$
$$y_i(\vec{x}_i \cdot \vec{w} + b) - 1 = 0 \text{ for samples in 'gutter'}$$



SUPPORT VECTOR MACHINES



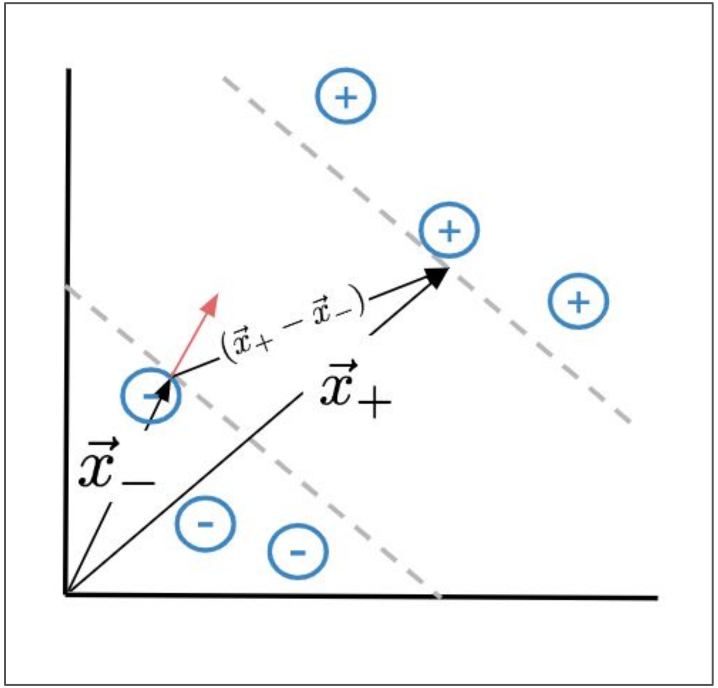
$$\text{Width} = (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{||w||}$$

$$(\vec{x}_+ \cdot \vec{w} + b) - 1 = 0$$
$$(\vec{x}_+ \cdot \vec{w} + b) = 1$$
$$\vec{x}_+ \cdot \vec{w} = 1 - b$$

$$-(\vec{x}_- \cdot \vec{w} + b) - 1 = 0$$
$$(\vec{x}_- \cdot \vec{w} + b) = -1$$
$$\vec{x}_- \cdot \vec{w} = -1 - b$$

$$\text{Width} = \frac{(1-b)-(-1-b)}{||w||}$$
$$\text{Width} = \frac{1-b+1+b}{||w||} = \frac{2}{||w||}$$

SUPPORT VECTOR MACHINES



$$\text{Width} = (\vec{x}_+ - \vec{x}_-) \cdot \frac{\vec{w}}{||w||} = \frac{2}{||w||}$$

For computational convenience take: $min(||w||) \rightarrow min(\frac{1}{2}||\vec{w}||^2)$

Solve using Lagrange multiplier $L = f(x) - \lambda g(x)$

$$L = \frac{1}{2}||\vec{w}||^2 - \sum_{i=1}^{n} \alpha_i [y_i(\vec{w} \cdot \vec{x}_i + b) - 1]$$

$$\frac{\partial L}{\partial \vec{w}} = \vec{w} - \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i = 0$$

$$\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i$$

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i \vec{x}_i = 0$$

SUPPORT VECTOR MACHINES

$$L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j)$$

Optimization depends only on a pair of samples!

Original Decision Rule: $\vec{w} \cdot \vec{u} + b \geq 0$

(recall $\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i$)

New Decision Rule: If $\sum \alpha_i y_i \vec{x}_i \cdot \vec{u} + b \geq 0$ then the sample is positive

What if the data isn't linearly separable?

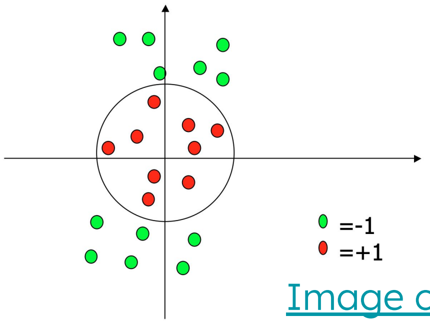
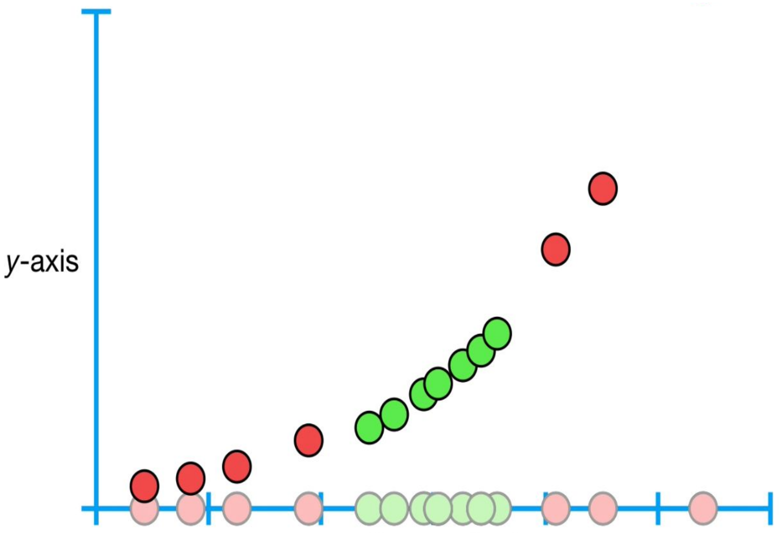


Image credit

SUPPORT VECTOR MACHINES

We can transform the data to a higher dimension:



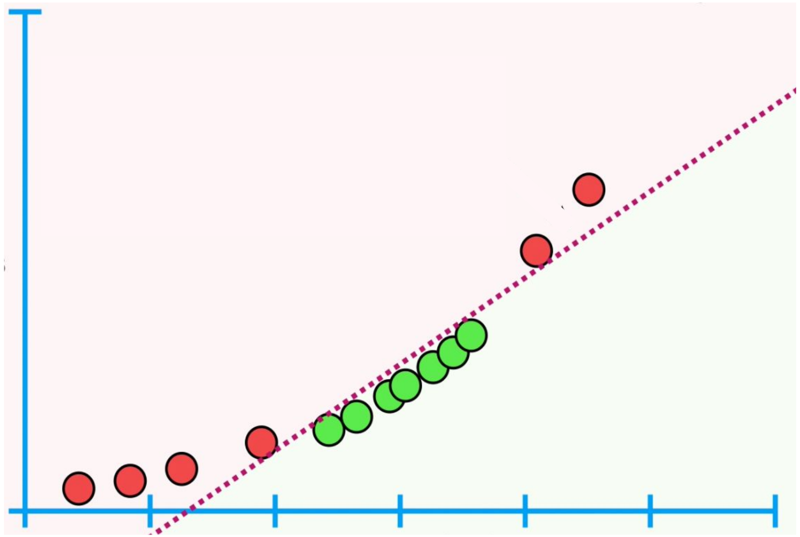
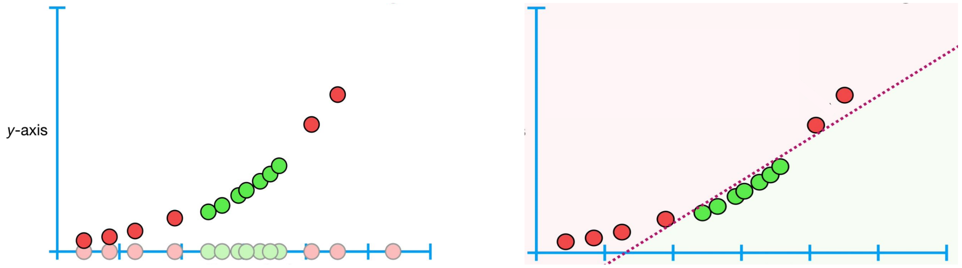


Image credit

How do we know how to transform the data?

SUPPORT VECTOR MACHINES



Kernel Functions - Systematically find support vector classifiers in higher dimensions. Transform data with φ, we can replace internal dot product $(x_i \bullet x_j)$ with $K(x_i, x_j) = \phi(x_i) \bullet \phi(x_j)$. The function we want to optimize becomes:

$$L = \sum a_i - \tfrac{1}{2} \sum a_i a_j y_i y_j K(x_i \bullet x_j)$$

Kernel functions calculate relationships between every pair of points as if they're in the higher dimension, without actually doing the transformation. This is called the **kernel trick**.

SUPPORT VECTOR MACHINES

Radial Basis Function (RBF):

- Works in infinite dimensions (so impossible to visualize) but behaves similar to a weighted nearest neighbor model on new data points

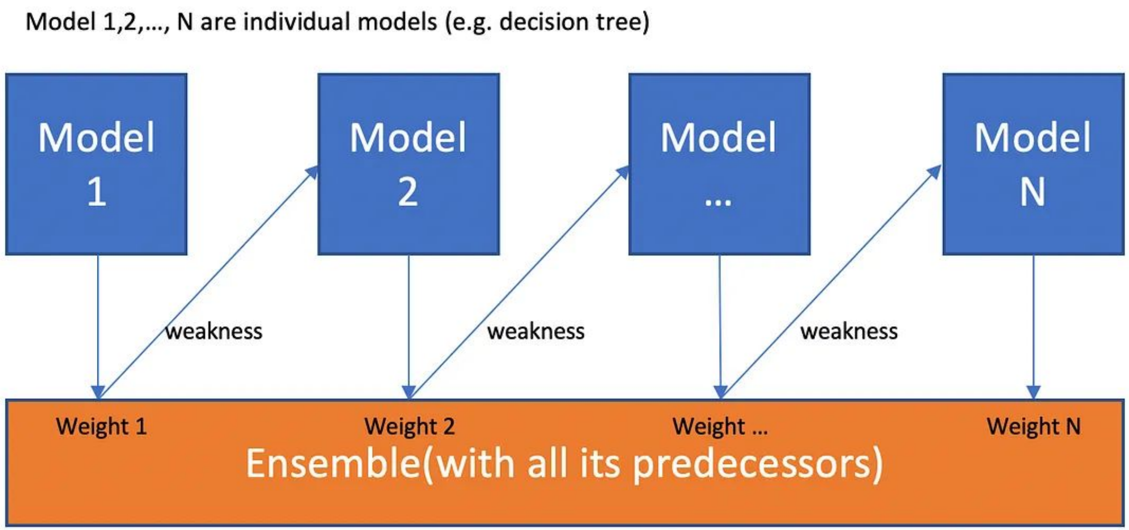
$$\mathrm{RBF} = e^{-\gamma ||a-b||^2}$$

- a and b are input points
- Gamma is determined by cross-validation scales the squared Euclidean norm.

BOOSTING

GOAL — Support/work alongside models by correcting errors or “weaknesses” within previous models

We choose to use a weighted majority vote



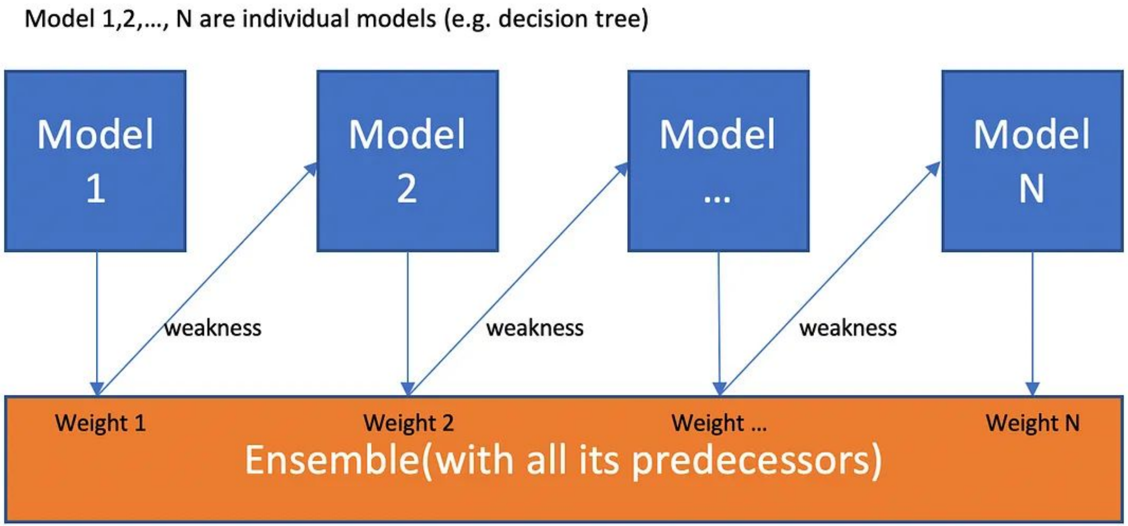
BOOSTING

Let f(x) denote the predicted label of the boosted classifier.

$f_k(x)$ represents the predicted label of the kth model

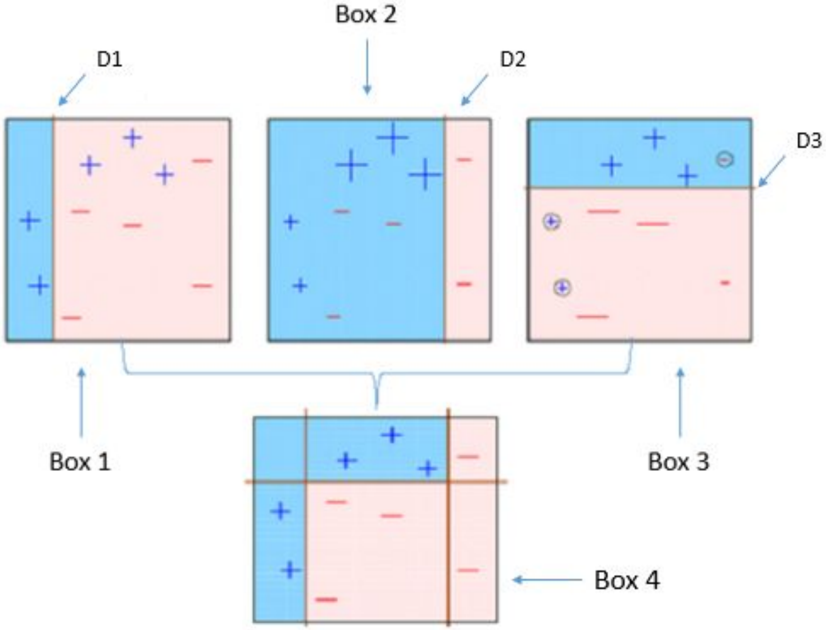
a_k represents the performance of the kth model

$$f(x) = \text{sign}(\sum_{i=1}^{K} \alpha_k f_k(x))$$



BOOSTING (AdaBoost: 1995)

Idea: Give larger weights to points not classified by previous models + smaller weights to points classified by previous model



$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha t} & \text{if } f_t(x_i) = y_i \\ e^{\alpha t} & \text{if } f_t(x_i) \neq y_i \end{cases}$$

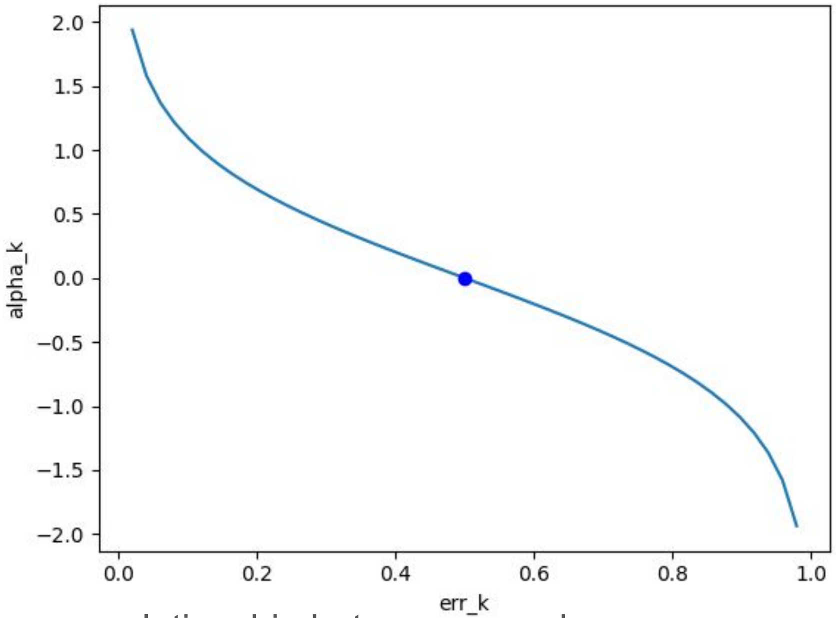
BOOSTING (AdaBoost)

Overall classifier is a weighted majority vote, with the weight of the kth classifier depending on its performance a_k.

$$f(x) = \mathrm{sign}(\sum_{i=1}^{K} \alpha_k f_k(x))$$

$$err_k = \frac{\sum_{i=1}^{n} w_i \mathbb{1}\{y_i \neq f_k(x_i)\}}{\sum_{i=1}^{n} w_i}$$

$$\alpha_k = \frac{1}{2} \log(\frac{1 - err_k}{err_k})$$



relationship between a_k and err_k

ADABOOST AS MINIMIZER OF EXPONENTIAL LOSS

AdaBoost is a Greedy algorithm!

Exponential loss formula: $L_{exp}(\mathbf{x}, y) = e^{-yf(\mathbf{x})}$

Recall formulation of boosting classifier f(x) $f(x) = \text{sign}(\sum_{i=1}^{K} \alpha_k f_k(x))$

Plugging this in, let us define the loss function of the boosting classifier as E

$$E = \sum_i e^{-y_i \sum_{i=1}^{K} \alpha_k f_k(x_i)}$$

ADABOOST AS MINIMIZER OF EXPONENTIAL LOSS

$$E = \sum_i e^{-y_i \sum_{i=1}^{K} \alpha_k f_k(x_i)}$$

Exponential loss definition

$$= \sum_i e^{-y_i \sum_{i=1}^{K-1} \alpha_k f_k(x_i) - y_i \alpha_K f_K(x_i)}$$

Split into first K-1, Kth classifiers

$$= \sum_i e^{-y_i \sum_{i=1}^{K-1} \alpha_k f_k(x_i)} e^{-y_i \alpha_K f_K(x_i))}$$

$$= \sum_i w_i^{(k)} e^{-y_i \alpha_K f_K(x_i))}$$

Treat first K-1 terms as constant w_i

ADABOOST AS MINIMIZER OF EXPONENTIAL LOSS

$$= \sum_{i:f_k(x_i)=y_i} w_i^{(k)} e^{-\alpha_k} + \sum_{i:f_k(x_i)\neq y_i} w_i^{(k)} e^{\alpha_k}$$

Split into matching and unmatching cases

$$= \sum_{i:f_k(x_i)=y_i} w_i^{(k)} e^{-\alpha_k} + \sum_{i:f_k(x_i)\neq y_i} w_i^{(k)} e^{-\alpha_k} - \sum_{i:f_k(x_i)\neq y_i} w_i^{(k)} e^{-\alpha_k} + \sum_{i:f_k(x_i)\neq y_i} w_i^{(k)} e^{\alpha_k}$$

$$= \sum_{i} w_i^{(k)} e^{-\alpha_k} + \sum_{i:f_k(x_i)\neq y_i} w_i^{(k)} (e^{\alpha_k} - e^{-\alpha_k})$$

$$= e^{-\alpha_k} \sum_{i} w_i^{(k)} + (e^{\alpha_k} - e^{-\alpha_k}) \sum_{i} w_i^{(k)} \mathbb{1}\{y_i \neq f_k(x_i)\}$$

ADABOOST AS MINIMIZER OF EXPONENTIAL LOSS

Exponential loss of Boosting (E): $= e^{-\alpha_k} \sum_i w_i^{(k)} + (e^{\alpha_k} - e^{-\alpha_k}) \sum_i w_i^{(k)} \mathbb{1}\{y_i \neq f_k(x_i)\}$

Now, find the minima of E w.r.t alpha by taking the first derivative

$$\frac{dE}{d\alpha_k} = -\alpha_k e^{-\alpha_k} \sum_i w_i^{(k)} + \alpha_k (e^{\alpha_k} - e^{-\alpha_k}) \sum_i w_i^{(k)} \mathbb{1}\{y_i \neq f_k(x_i)\} = 0$$

Divide both sides by $\frac{\alpha_k}{\sum_i w_i^{(m)}}$

$$0 = -e^{-\alpha_k} + e^{\alpha_k}\epsilon_k - e^{-\alpha_k}\epsilon_k$$

$$e^{\alpha_k}\epsilon_k = e^{-\alpha_k}(1 - \epsilon_k)$$

$$\alpha_k + \ln \epsilon_k = -\alpha_k + \ln(1 - \epsilon_k)$$

$$2\alpha_k = ln(\frac{1 - \epsilon_k}{\epsilon_k})$$

$$\alpha_k = \frac{1}{2} ln(\frac{1 - \epsilon_k}{\epsilon_k})$$

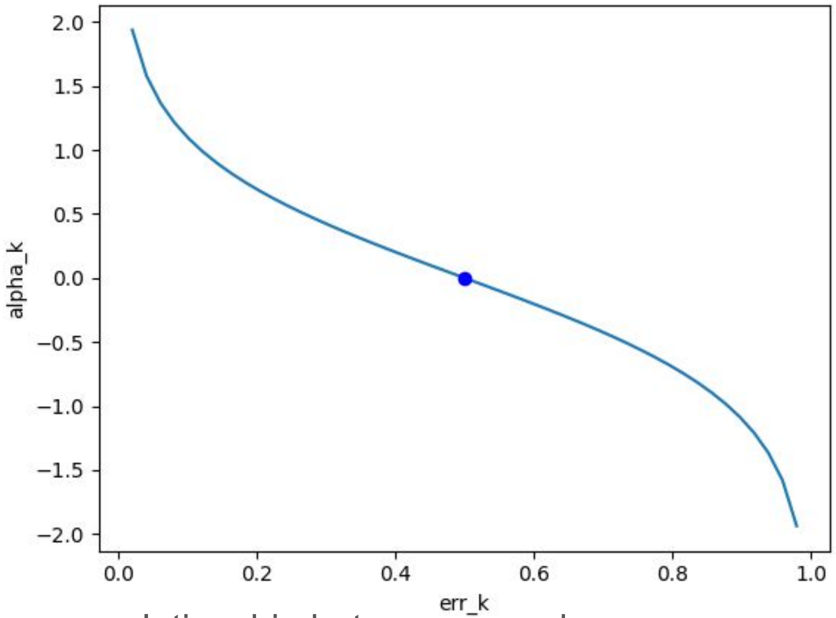
BOOSTING (AdaBoost)

Overall classifier is a weighted majority vote, with the weight of the kth classifier depending on its performance a_k.

$$f(x) = \text{sign}(\sum_{i=1}^{K} \alpha_k f_k(x))$$

$$err_k = \frac{\sum_{i=1}^{n} w_i \mathbb{1}\{y_i \neq f_k(x_i)\}}{\sum_{i=1}^{n} w_i}$$

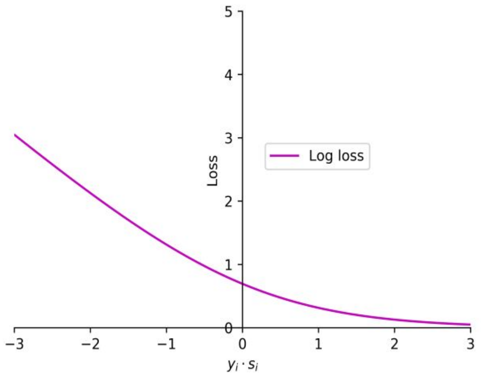
$$\boxed{\alpha_k = \frac{1}{2}\log(\frac{1 - err_k}{err_k})}$$



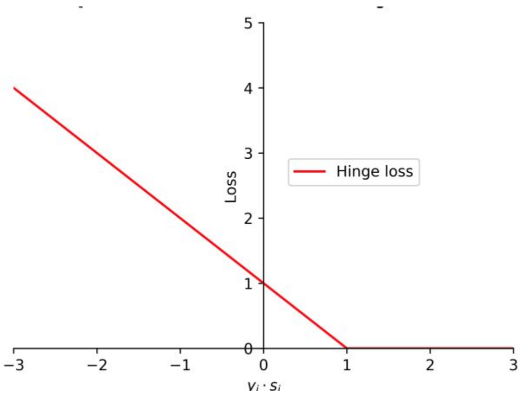
relationship between a_k and err_k

COMPARISON OF LOSS FUNCTIONS

Logistic Regression:
Cross-entropy



SVM:
Hinge loss



Boosting (Adaboost):
Exponential loss

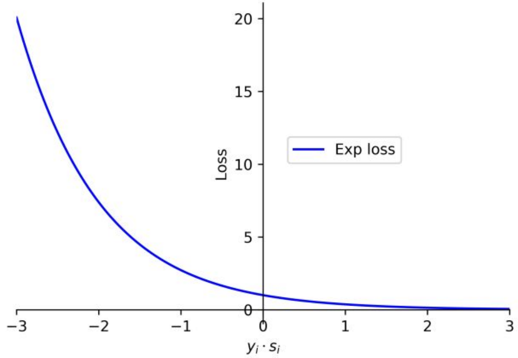


Image credit

COMPARISON OF LOSS FUNCTIONS

Both logistic regression and SVM look to draw a decision boundary, but logistic regression also calculates probability of a classification.

SVM doesn't consider probability, but is able to better handle outliers and nonlinearity.

Log loss is similar to hinge loss but is a smooth function (can be optimized with the gradient descent method)

While log loss grows slowly for negative values, exponential loss is more aggressive.

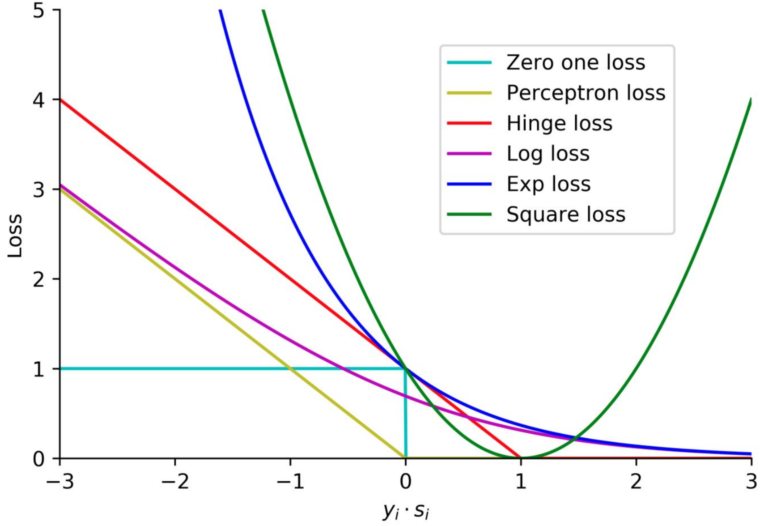


Image credit

LYRICS BASED RECOMMENDATIONS -- LLMs

Goal: generate recommendations based on content of lyrics, attempt to understand lyrics using Large Language Models (LLMs)

- Large Language Models - advanced machine learning models trained on textual data, aim to understand human language
- Co-occurrence matrices can be used to learn relationships between words (ex. “fast” vs. “rapid” vs. “speed”)
    - Measure how often words appear together or are used interchangeably
    - However - analyzing pairwise relationships between all words results in extremely large, sparse matrices
- Word embeddings - representations of words in a low-dimensional, dense vector space

WORD EMBEDDINGS (TF*IDF: 1972)

Bag-of-words (BOW) - store a text corpus as a vector of words and corresponding frequencies (unordered text representation)

- TF * IDF = term frequency x inverse document frequency
  - Term Frequency - how often does the term t appear?

$$tf_{t,d} = count(t, d)$$

  - Inverse Document Frequency - weight of each word, inverse to how often the word appears (common words will be weighted less overall)

$$idf_{t,D} = \log \frac{|D|}{|\{d \in D, count(t, d) > 0\}|}$$

  - *t* - term, *d* - context, */D/* - entire document/corpus
  - Similar to the BOW method, a text corpus is represented by a vector of words and the TF*IDF frequency

WORD EMBEDDINGS (Word2Vec: 2013)

- TF-IDF cannot account for similarities among words
- Word2Vec uses neural networks to learn word associations (synonymy, lexical substitution) and generate word embeddings
- W2V models produces similar word embeddings for words that are used in the same context
- Uses an unordered Continuous Bag-of-words approach to produce word embeddings

WORD EMBEDDINGS (BERT: 2018)

- BERT - Bidirectional Encoder Representations from Transformers
- Improves upon directional models (read left-to-right or right-to-left) by using bidirectional training to understand the entire context of a word
- Using a [MASK] token for each word in a sequence, BERT understands the word by predicting the original value based on the context

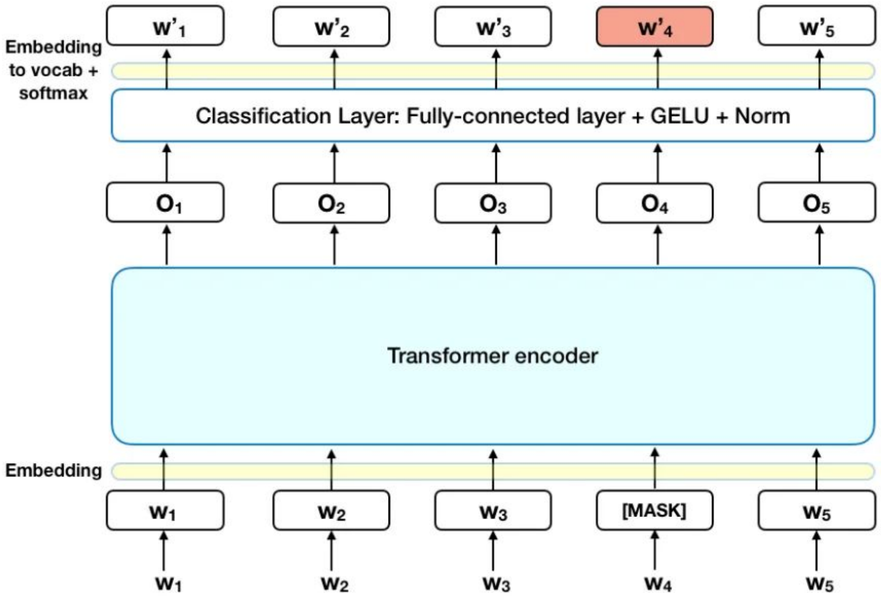


Image Credit: BERT Explained

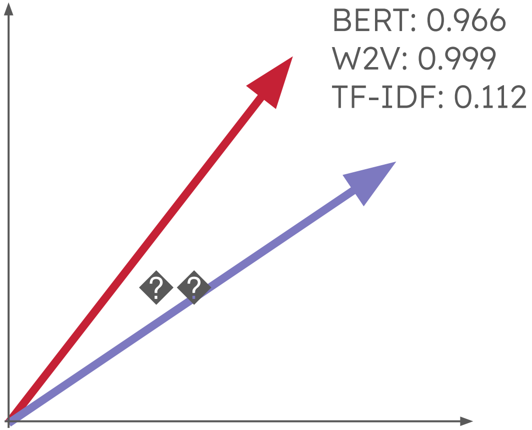
COSINE SIMILARITY

- Using word embeddings (vectorized representation of text), we can calculate the similarity between lyrics vectors using cosine similarity

“you broke me first” - Tate McRae

“good 4 u” - Olivia Rodrigo

Generate word embeddings using BERT, TF-IDF, or Word2Vec



$$\text{Cosine Sim}(A,B) = \frac{A \cdot B}{||A|| \times ||B||} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

DEMO

Google Colab Link

RESULTS — Emily

	track_name	artist(s)_name	score
0	Iris	The Goo Goo Dolls	1
1	Labyrinth	Taylor Swift	1
2	Mr. Brightside	The Killer	1
3	Do I Wanna Know?	Arctic Monkeys	1
4	Everybody Wants To Rule The World	Tears For Fears	1
5	Dance Monkey	Tones and I	-1
6	If We Ever Broke Up	Mae Stephens	-1
7	Light Switch	Charlie Puth	-1
8	Bad Habits	Ed Sheeran	-1
9	Ghost	Justin Bieber	-1

Input: 5 likes, 5 dislikes
Likes assigned score(1)
Dislikes assigned score(-1)

Logistic Regression

	track_name	artist(s)_name	pred
952	Alone	Burna Boy	1
399	TV	Billie Eilish	1
731	Fuera del mercado	Danny Ocean	1
732	X Ultima Vez	Daddy Yankee, Bad Bunny	1
395	Space Song	Beach House	1

SVMs

	track_name	artist(s)_name	pred
952	Alone	Burna Boy	1
243	Unstoppable	Sia	1
734	In My Head	Lil Tjay	1
388	STAR WALKIN' (League of Legends Worlds Anthem)	Lil Nas X	1
387	Lift Me Up - From Black Panther: Wakanda Forev...	Rihanna	1

Boosting

	track_name	artist(s)_name	pred
952	Alone	Burna Boy	1.0
379	Devil Don't Know	Morgan Wallen	1.0
726	O.O	NMIXX	1.0
395	Space Song	Beach House	1.0
394	Escapism. - Sped Up	RAYE, 070 Shake	1.0

RESULTS — Emily

	track_name	artist(s)_name	score
0	Iris	The Goo Goo Dolls	1
1	Labyrinth	Taylor Swift	1
2	Mr. Brightside	The Killer	1
3	Do I Wanna Know?	Arctic Monkeys	1
4	Everybody Wants To Rule The World	Tears For Fears	1
5	Dance Monkey	Tones and I	-1
6	If We Ever Broke Up	Mae Stephens	-1
7	Light Switch	Charlie Puth	-1
8	Bad Habits	Ed Sheeran	-1
9	Ghost	Justin Bieber	-1

Input: 5 likes, 5 dislikes
Likes assigned score(1)
Dislikes assigned score(-1)

TF - IDF

	track_name	artist(s)_name	score
750	Falling	Harry Styles	0.564007
715	this is what falling in love feels like	JVKE	0.463133
357	Thought You Should Know	Morgan Wallen	0.198366
878	die first	Nessa Barrett	0.139239
550	Smokin Out The Window	Bruno Mars, Anderson .Paak, Silk Sonic	0.132895

W2V

	track_name	artist(s)_name	score
120	LUNA	Junior H, Peso Pluma	-0.366168
222	Ch y la Pizza	Fuerza Regida, Natanael Cano	-0.370910
306	La Bebe	Yng Lvcas	-0.373553
190	Bebe Dame	Fuerza Regida, Grupo Frontera	-0.373991
9	La Bebe - Remix	Peso Pluma, Yng Lvcas	-0.375782

BERT

	track_name	artist(s)_name	score
395	Space Song	Beach House	-0.088053
846	Keep Driving	Harry Styles	-0.092675
120	LUNA	Junior H, Peso Pluma	-0.221548
139	Romantic Homicide	d4vd	-0.232461
910	The Scientist	Coldplay	-0.233431

RESULTS — Liz

	track_name	artist(s)_name	score
0	Take Me To Church	Hozier	1
1	august	Taylor Swift	1
2	Matilda	Harry Styles	1
3	Easy On Me	Adele	1
4	Let Me Down Slowly	Alec Benjamin	1
5	golden hour	JVKE	-1
6	Unholy (feat. Kim Petras)	Sam Smith, Kim Petras	-1
7	Unstoppable	Sia	-1
8	Bad Habits	Ed Sheeran	-1
9	Made You Look	Meghan Trainor	-1

Input: 5 likes, 5 dislikes
Likes assigned score(1)
Dislikes assigned score(-1)

Logistic Regression

track_name	artist(s)_name	pred
All Of The Girls You Loved Before	Taylor Swift	1
Closer	The Chainsmokers, Halsey	1
Chale	Eden Muï¿½ï	1
DARARI	Treasure	1
this is what falling in love feels like	JVKE	1

SVMs

track_name	artist(s)_name	pred
All Of The Girls You Loved Before	Taylor Swift	1
I Was Never There	The Weeknd, Gesaffelstein	1
I'm Tired - From "Euphoria" An Original HBO Se...	Labrinth	1
Chale	Eden Muï¿½ï	1
DARARI	Treasure	1

Boosting

track_name	artist(s)_name	pred
I Hate U	SZA	1.0
Flowers	Lauren Spencer Smith	1.0
San Lucas	Kevin Kaarl	1.0
With you	HA SUNG WOON, Jimin	1.0
Talking To The Moon	Bruno Mars	1.0

RESULTS — Liz

	track_name	artist(s)_name	score
0	Take Me To Church	Hozier	1
1	august	Taylor Swift	1
2	Matilda	Harry Styles	1
3	Easy On Me	Adele	1
4	Let Me Down Slowly	Alec Benjamin	1
5	golden hour	JVKE	-1
6	Unholy (feat. Kim Petras)	Sam Smith, Kim Petras	-1
7	Unstoppable	Sia	-1
8	Bad Habits	Ed Sheeran	-1
9	Made You Look	Meghan Trainor	-1

Input: 5 likes, 5 dislikes
Likes assigned score(1)
Dislikes assigned score(-1)

TF - IDF

	track_name	artist(s)_name	score
324	Say You Won't Let Go	James Arthur	0.624978
322	I Love You So	The Walters	0.598729
588	happier	Olivia Rodrigo	0.484417
655	City of Gods	Kanye West, Alicia Keys, Fivio Foreign	0.480291
102	Chemical	Post Malone	0.461277

W2V

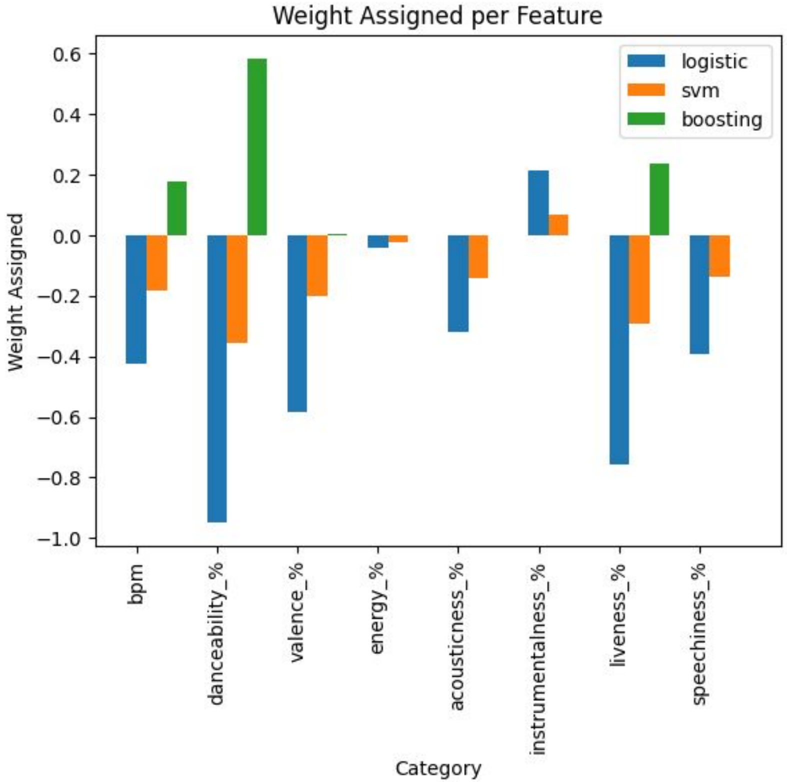
	track_name	artist(s)_name	score
851	Daydreaming	Harry Styles	1.002640
148	Those Eyes	New West	1.002502
593	Rolling in the Deep	Adele	1.002334
403	One Kiss (with Dua Lipa)	Calvin Harris, Dua Lipa	1.002098
83	Back To December (Taylor's Version)	Taylor Swift	1.001916

BERT

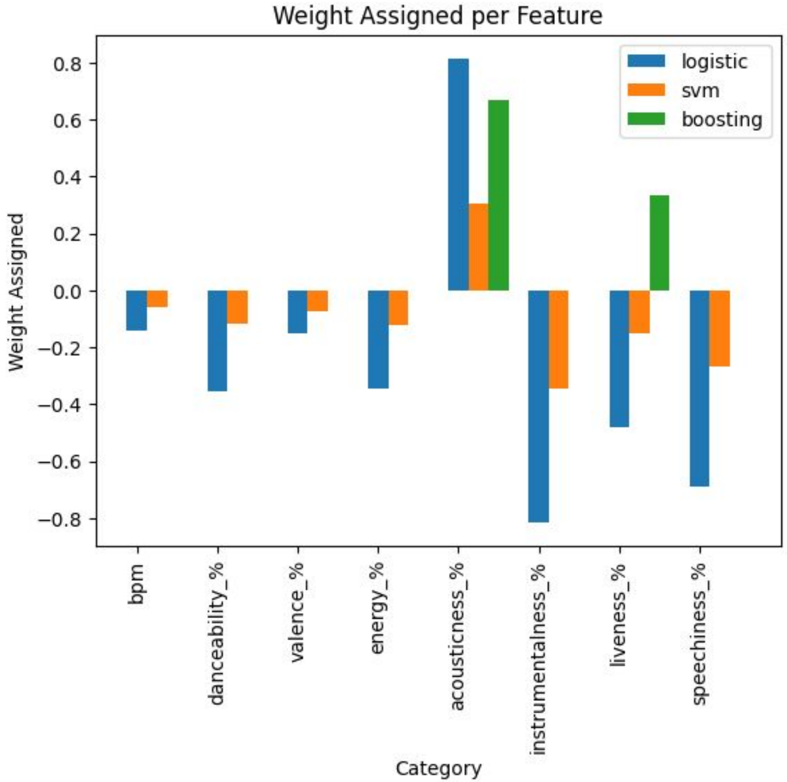
	track_name	artist(s)_name	score
900	Forget Me	Lewis Capaldi	0.771068
255	Curtains	Ed Sheeran	0.748584
113	Mine (Taylor's Version)	Taylor Swift	0.748451
83	Back To December (Taylor's Version)	Taylor Swift	0.746609
513	good 4 u	Olivia Rodrigo	0.737577

RESULTS

Emily's Model Weighting



Liz's Model Weighting



SPOTIFY RECOMMENDATION METHODOLOGY

1. Artist-sourced metadata: collected through Spotify for Artists (S4A)
    - Artists label songs with title, featured artists, release date, etc.
    - Includes genre and sub-genre tags, music culture tags, mood tags, instruments used, etc. in addition to basic details
2. Raw audio signal analysis
    - Objective audio attributes such as **instrumentalness** - given a score between 0 and 1 depending on the amount of vocals in the track
    - Subjective audio attributes such as **danceability, energy,** and **valence**
3. Text analysis with Natural Language Processing
    - Lyrics analysis
    - Web-crawled data analysis
    - User-generated playlist analysis

CHALLENGES / FUTURE OF THE FIELD

- Promoting new artists - both collaborative and content based filtering algorithms struggle with promoting new artists
  - Content-based features such as popularity / # of streams may prevent songs from being recommended
- Cultural bias (within datasets and NLP algorithms)
  - Limitations of Word2Vec
- General field shift towards mood based listening
  - Can help people lower stress to listen to “happy” music (from study in COVID-19)
  - Streaming platforms/accounts push music personalization
  - Music is becoming (relatively) less of a product and more of a personal experience → high demand for personalized collections and refined recommendations

OUR FUTURES

Liz: • MS in Applied Math or Statistics • Interested in actuarial science • Want to keep playing and discovering new music!	Emily: • Will work in data analytics post grad in financial service • Want to keep discovering new music and ways to find it!
Ananya: • Interested in working with more ML projects • Considering MS in CS!	Emilie: • Planning to continue to explore more music • Hoping to work with more stats + ML in the future

Thank you!

Special thanks to Professor Wiggins and Brian Whitman for their insight

ALL REFERENCES (Slide 1)

- Agrawal, Naman. “Decoding Logistic Regression Using MLE.” Analytics Vidhya. Published as part of Data Science Blogathon. March 22, 2022. https://www.analyticsvidhya.com/blog/2022/02/decoding-logistic-regression-using-mle/
- Ansaf, Salleb-Aouissi. “Machine Learning Ensemble Methods”. Artificial Intelligence COMS4701. Fall 2023. Columbia University.
- Belyadi, H., and Haghighat, A. “Chapter 5 - Supervised learning”. Machine Learning Guide for Oil and Gas Using Python. Gulf Professional Publishing. 2021. Pages 169-295, ISBN 9780128219294. https://doi.org/10.1016/B978-0-12-821929-4.00004-4
- Bertin-Mahieux, T., Ellis, D., Whitman, B., and Lamere, P. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011. http://millionsongdataset.com/
- Berwick, Robert. “An Idiot’s guide to Support vector machines (SVMs)”. MIT 6.034 AI. Fall 2011. Massachusetts Institute of Technology. https://web.mit.edu/6.034/wwwbob/svm.pdf
- Brooks-Bartlett, Jonny. “Probability concepts explained: Maximum likelihood estimation”. Towards Data Science. January 3, 2018. https://towardsdatascience.com/probability-concepts-explained-maximum-likelihood-estimation-c7b4342fdbb1
- Brubaker, Marcus. “18 AdaBoost”. CSC 411: Introduction to Machine Learning and Data Mining. Fall 2015. University of Toronto. https://www.cs.toronto.edu/~mbrubake/teaching/C11/Handouts/AdaBoost.pdf
- Chakraborty, Arunava. “Derivative of the Sigmoid function”. Published by Arc in Towards Data Science. July 7, 2018. https://towardsdatascience.com/derivative-of-the-sigmoid-function-536880cf918e
- Elgiriyewithana, Nidula. “Most Streamed Spotify Songs 2023” Version 1. August, 2023. Retrieved November 10th from https://www.kaggle.com/datasets/nelgiriyewithana/top-spotify-songs-2023/data
- Feneberg AC, Stijovic A, Forbes PAG, et al. Perceptions of Stress and Mood Associated With Listening to Music in Daily Life During the COVID-19 Lockdown. JAMA Netw Open. 2023;6(1):e2250382. doi:10.1001/jamanetworkopen.2022.50382 https://pubmed.ncbi.nlm.nih.gov/36626171/
- Grosse, Roger. “Support Vector Machines and Boosting”. Machine Learning CSC 2515. Fall 2019. University of Toronto. https://www.cs.toronto.edu/~rgrosse/courses/csc2515_2019/readings/SVM-and-boosting.pdf
- “Logistic Regression with Maximum Likelihood”. Youtube, uploaded by Endless Engineering, January 6, 2019. https://www.youtube.com/watch?v=TM1lijyQnaI
- Mahedero, Jose & Martinez, Alvaro & Cano, Pedro & Koppenberger, Markus & Gouyon, Fabien. (2005). Natural language processing of lyrics. 475-478. 10.1145/1101149.1101255. https://www.researchgate.net/publication/221573745_Natural_language_processing_of_lyrics
- Pappu, Vijay. “Lecture 3: Linear models for regression, Logistic Regression”. Applied Machine Learning W4995. Fall 2023. Columbia University. https://drive.google.com/file/d/1pKInzoPMyygDQP-
- Pastukhov, Dmitry. “Inside Spotify’s Recommender System: A Complete Guide to Spotify Recommendation Algorithms”. Music Tomorrow. February 9, 2022. https://www.music-tomorrow.com/blog/how-spotify-recommendation-system-works-a-complete-guide-2022

ALL REFERENCES (Slide 2)

- Piech, Chris. “Logistic Regression”. Probability for Computer Scientists CS109. May 20, 2016. Stanford University. https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/pdfs/40%20LogisticRegression.pdf
- Schapire, R.E. and Freund, Y.. “Boosting: Foundations and Algorithms,” 2012. The MIT Press. ISBN (electronic): 9780262301183. https://doi.org/10.7551/mitpress/8291.001.0001
- Shalizi, Cosma. “Logistic Regression”. Undergraduate Advanced Data Analysis 36-402. Spring 2012. Carnegie Mellon University. https://www.stat.cmu.edu/~cshalizi/uADA/12/lectures/ch12.pdf
- Starmer, Josh. “Logistic Regression Details Pt 1: Coefficients.” YouTube, uploaded by StatQuest with Josh Starmer, June 4, 2018. https://www.youtube.com/watch?v=vN5cNN2-HWE&t=621s
- Starmer, Josh. “Logistic Regression Details Pt 2: Maximum Likelihood.” YouTube, uploaded by StatQuest with Josh Starmer, June 11, 2018. https://www.youtube.com/watch?v=BfKanl1aSG0&list=PLblh5JKOoLUKxzEP5HA2d-Li7IJkHfXSe&index=5
- Starmer, Josh. “StatQuest: Logistic Regression.” YouTube, uploaded by StatQuest with Josh Starmer, March 5, 2018. https://www.youtube.com/watch?v=yIYKR4sgzI8
- Swaminathan, Saishruthi. “Logistic Regression - Detailed Overview.” Towards Data Science. March 15, 2018. https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc
- Tian, H. Cai, H., Wen, J., Li, S. and Li, Y., "A Music Recommendation System Based on logistic regression and eXtreme Gradient Boosting," 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 2019, pp. 1-6, doi: 10.1109/IJCNN.2019.8852094. https://ieeexplore.ieee.org/abstract/document/8852094
- Varma, Rohan. “Picking Loss Functions - A comparison between MSE, Cross Entropy, and Hinge Loss”. Rohan Varma. January 9, 2018. https://rohanvarma.me/Loss-Functions/
- Wagh, Shruti. “Logistic Regression: Understanding odd and log-odds”. Medium. Published in Women in Collaborative Data Science. December 28, 2020. https://medium.com/wicds/logistic-regression-understanding-odds-and-log-odds-61aecdc88846
- “What Is a Recommendation System?” NVIDIA Data Science Glossary, NVIDIA Corporation, 2023, www.nvidia.com/en-us/glossary/data-science/recommendation-system/#:~:text=A%20recommendation%20system%20is%20an,demographic%20information%2C%20and%20other%20factors.
- Yan, Lisa. “Maximum Likelihood Estimation”. Probability for Computer Scientists CS109. May 20, 2020. Stanford University. https://web.stanford.edu/class/cs109/
- “6.1 Introduction to GLMs”. Analysis of Discrete Data STAT 504. Penn State Eberly College of Science. https://online.stat.psu.edu/stat504/lesson/13-0