CUKUROVA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CEN481 - INTRODUCTION TO DATA MINING

2019556055 - ALİ CAN SARIBOĞA
2019556032 - OĞUZHAN GÜNEY
2019556006 - ANIL ALKIŞ
2019556039 - UTKU KAYA
2018556014 - AHMET HAKAN BOZ

DATASET - TURKISH MUSIC EMOTION

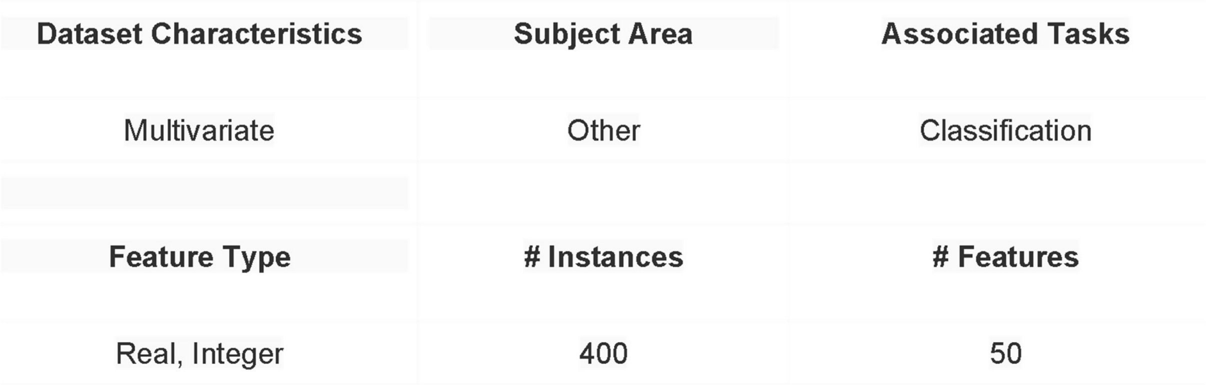
FALL
2023 - 2024

TURKISH MUSIC EMOTION

Introduction

The aim is to analyze the emotional content in music signals. In this dataset, five different algorithms have been applied, resulting in five distinct models. In the results section, the obtained outcomes are compared.

General Information About Dataset:



Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Other	Classification
Feature Type	**# Instances**	**# Features**
Real, Integer	400	50

Dataset:

The Turkish Music Emotion Dataset prepared by Mehmet Bilal Er was used as the dataset.

The dataset is structured with a discrete model, comprising four classes: happy, sad, angry, and relax. To compile the dataset, both verbal and non-verbal music pieces were selected from various genres of Turkish music. A total of 100 music pieces were assigned to each class to ensure an equal number of samples in all categories. The original dataset consists of 400 samples, each lasting 30 seconds.

Number of Data in Each class: Relax = 100, Happy = 100, Sad = 100, Angry = 100.
Dataset Attributes:

RMSenergy_Mean: Represents the average root mean square energy of the sound. Values typically range between 0.010 and 0.431, reflecting the energy level of the sound.
Lowenergy_Mean: Represents the average of low-energy values. This value indicates how much low energy the sound generally has. Values range between 0.302 and 0.703.

Fluctuation_Mean: Represents the average of sound fluctuations. High fluctuation may indicate that the sound is intense or variable. The average value is 7.145932.

Tempo_Mean: Represents the average tempo of the sound. Values range between 48.284 and 195.026.

MFCC_Mean_1 - MFCC_Mean_13: Represents the average values of Mel-Frequency Cepstral Coefficients (MFCC), which represent different components of the sound's spectral features.

Roughness_Mean: Represents the average roughness features of the sound. A high roughness value may indicate that the sound is complex or harsh. The average value is 527.681365.

Roughness_Slope: Represents the average slope of roughness values. Values range between -0.525 and 0.584.

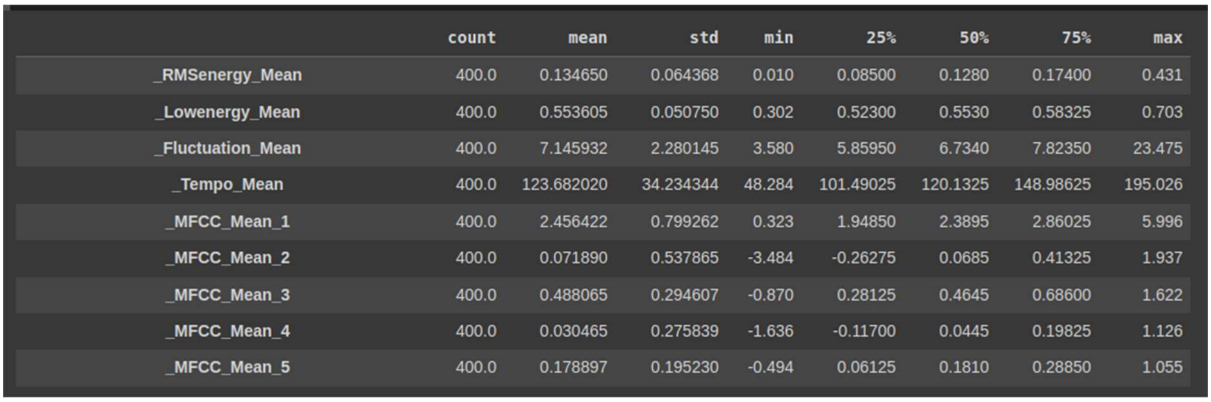
Zero-crossingrate_Mean: Represents the average zero-crossing rate of the sound, indicating the frequency of zero crossings in the sound wave.

AttackTime_Mean and AttackTime_Slope: Represent the average attack time and the average slope of attack time, respectively. Attack time reflects the onset speed of the sound.

Rolloff_Mean: Represents the average spectral rolloff, indicating how fast high frequencies diminish in the sound spectrum. Values range between 887.151 and 11508.298.

Eventdensity_Mean - HarmonicChangeDetectionFunction_PeriodEntropy: Represent the average values of different sound features, reflecting various characteristics of the sound.

A small excerpt of data related to the labels is provided below.

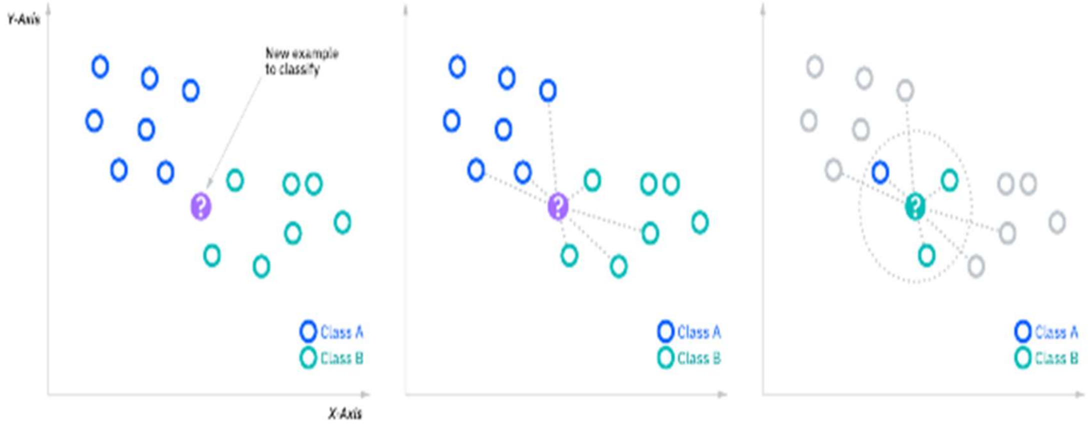


	count	mean	std	min	25%	50%	75%	max
_RMSenergy_Mean	400.0	0.134650	0.064368	0.010	0.08500	0.1280	0.17400	0.431
_Lowenergy_Mean	400.0	0.553605	0.050750	0.302	0.52300	0.5530	0.58325	0.703
_Fluctuation_Mean	400.0	7.145932	2.280145	3.580	5.85950	6.7340	7.82350	23.475
_Tempo_Mean	400.0	123.682020	34.234344	48.284	101.49025	120.1325	148.98625	195.026
_MFCC_Mean_1	400.0	2.456422	0.799262	0.323	1.94850	2.3895	2.86025	5.996
_MFCC_Mean_2	400.0	0.071890	0.537865	-3.484	-0.26275	0.0685	0.41325	1.937
_MFCC_Mean_3	400.0	0.488065	0.294607	-0.870	0.28125	0.4645	0.68600	1.622
_MFCC_Mean_4	400.0	0.030465	0.275839	-1.636	-0.11700	0.0445	0.19825	1.126
_MFCC_Mean_5	400.0	0.178897	0.195230	-0.494	0.06125	0.1810	0.28850	1.055

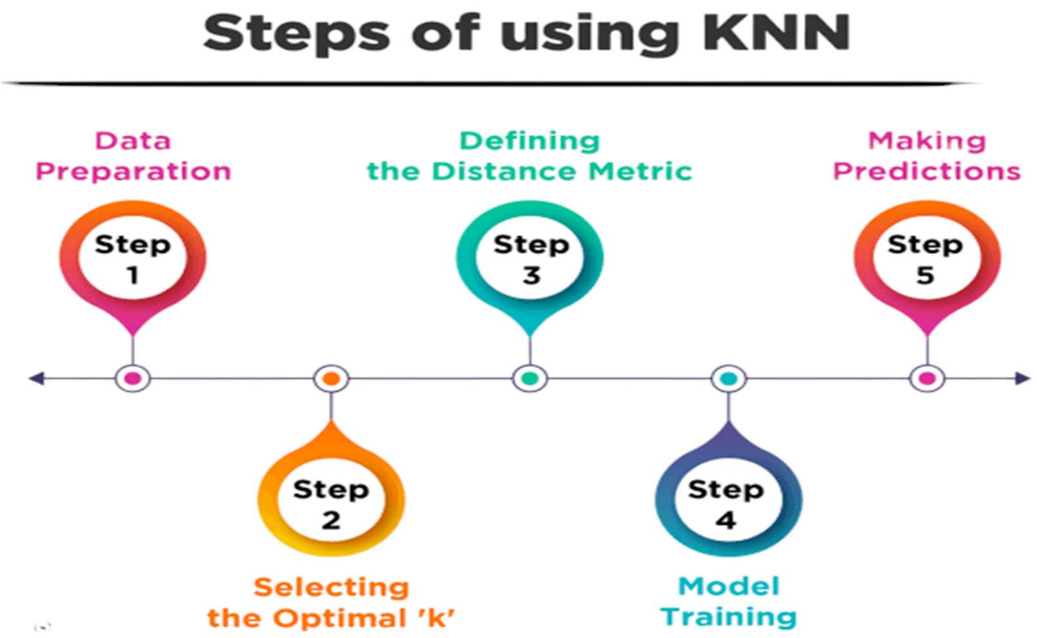
K-NEAREST NEIGHBOUR (KNN) ALGORITHM

General Description

Supervised learning is a subsection of machine learning generally associated with classification and regression based problems. Supervised learning implies that you are training a model using a labeled dataset. K Nearest Neighbours (KNN) falls under the supervised learning umbrella and is one of the core algorithms in machine learning. It’s a highly used, simple yet efficient example of a non-parametric, lazy learner classification algorithm.

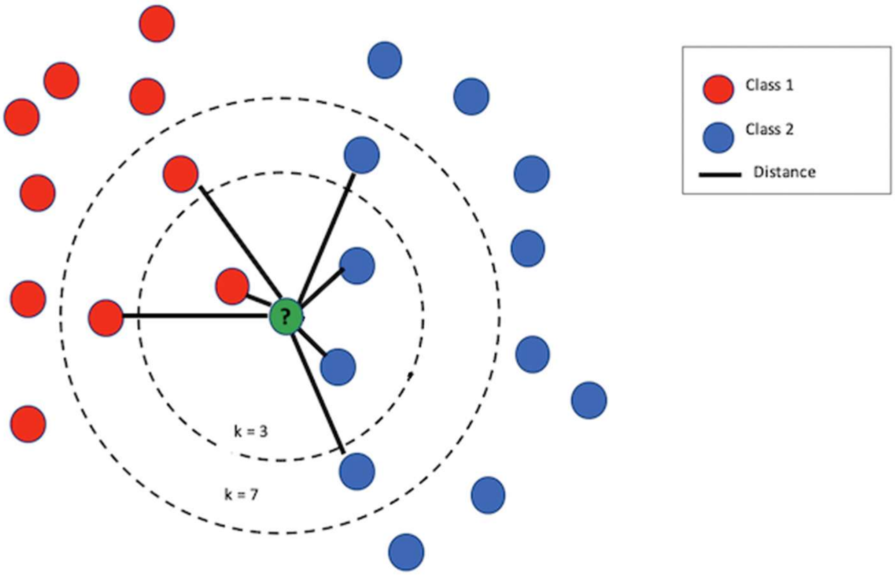


K- Nearest Neighbors or also known as K-NN belong to the family of supervised machine learning algorithms which means we use labeled (Target Variable) dataset to predict the class of new data point. The K-NN algorithm is a robust classifier which is often used as a benchmark for more complex classifiers such as Artificial Neural Network (ANN) or Support vector machine (SVM).



What is the KNN Algorithm?

The KNN algorithm classifies unclassified data points based on their proximity and similarity to other available data points. The underlying assumption this algorithm makes is that similar data points can be found near one another. It’s commonly used to solve problems in various industries because of its ease of use, application to classification and regression problems, and the ease of interpretability of the results it generates.



2 Important Parameters

The K-Nearest Neighbors (KNN) algorithm makes predictions based on two fundamental parameters: Distance and K value.

Distance: The algorithm calculates the distance between the point to be predicted and other points. Various distance calculation functions are used for this purpose. To give an example of a few of them; We can specify it as Euclidean, Minkowski, Manhattan, Hamming.

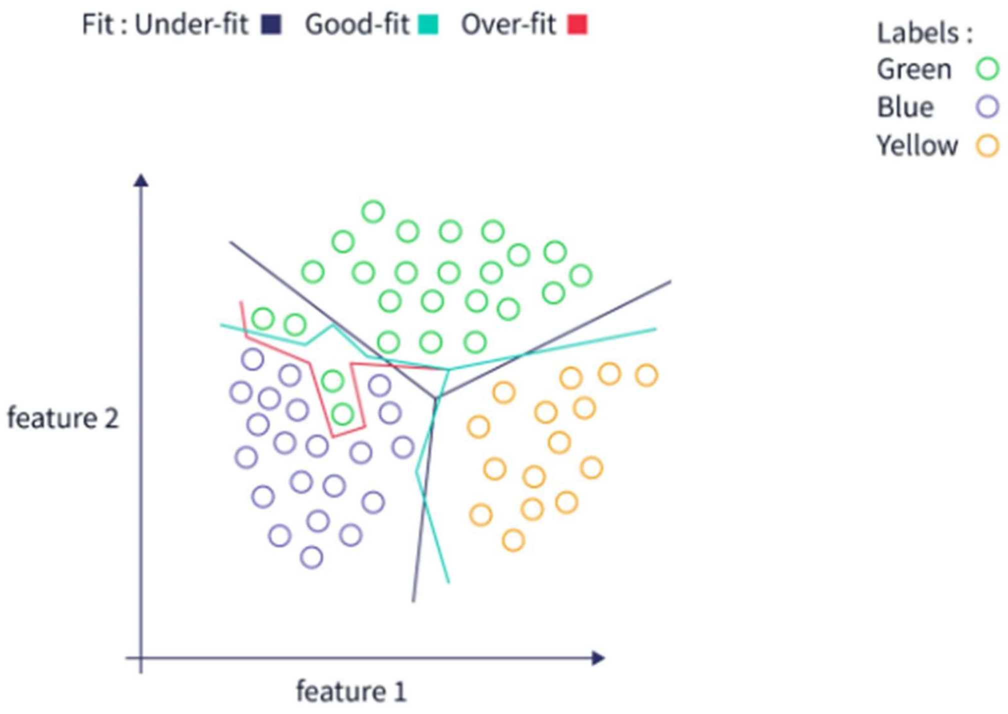
$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Euclid

$$\left(\sum_{i=1}^{n} \left|x_i - y_i\right|\right)^{1/p}$$

Manhattan

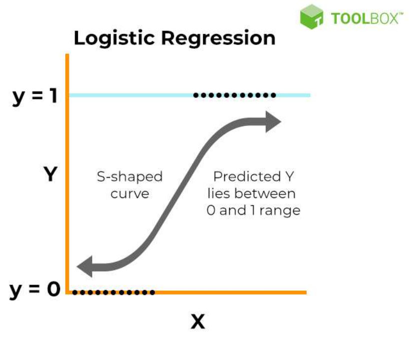
K (number of neighbors): We specify how many nearest neighbors will be considered for the calculation. The value of K directly affects the outcome. If K is 1, the likelihood of overfitting is very high. If K is too large, it may result in very general outcomes. Therefore, predicting the optimal value for K is the main challenge in the problem. Choosing an appropriate value of k in KNN is crucial to get the best out of the model. If the value of K is small, the model's error rate will be large, especially for new data points, since the number of votes is small. Hence, the model is overfitted and highly sensitive to noise in the input. Moreover, the boundaries for classification become too rigid, as seen in the image below.



Advantages and Disadvantages of KNN

ADVANTAGES	DISADVANTAGES
K-NN is pretty intuitive and simple.	K-NN is a slow algorithm.
K-NN has no assumptions.	Curse of Dimensionality.
No Training Step.	K-NN needs homogeneous features.
It constantly evolves.	Optimal number of neighbors.
Very easy to implement for multi-class problems.	Imbalanced data causes problems.
Can be used both for Classification and Regression.	Outlier sensitivity.
One Hyper Parameter (easier to add other hyper parameters).	Missing Value treatment.
Various distance criteria to choose.	

LOGISTIC REGRESSION ALGORITHM



This type of statistical model (also known as *logit model*) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following formulas:

Logit(pi) = 1/(1+ exp(-pi))

ln(pi/(1-pi)) = Beta_0 + Beta_1*X_1 + … + B_k*K_k

In this logistic regression equation, logit(pi) is the dependent or response variable and x is the independent variable. The beta parameter, or coefficient, in this model is commonly estimated via maximum likelihood estimation (MLE). This method tests different values of beta through multiple iterations to optimize for the best fit of log odds. All of these iterations produce the log likelihood function, and logistic regression seeks to maximize this function to find the best parameter estimate. Once the optimal coefficient (or coefficients if there is more than one independent variable) is found, the conditional probabilities for each observation can be calculated, logged, and summed together to yield a predicted probability. For binary classification, a probability less than .5 will predict 0 while a probability greater than 0 will predict 1. After the model has been computed, it's best practice to evaluate the how well the model predicts the dependent variable, which is called goodness of fit. The Hosmer–Lemeshow test is a popular method to assess model fit.

Interpreting logistic regression

Log odds can be difficult to make sense of within a logistic regression data analysis. As a result, exponentiating the beta estimates is common to transform the results into an odds ratio (OR), easing the interpretation of results. The OR represents the odds that an outcome will occur given a particular event, compared to the odds of the outcome occurring in the absence of that event. If the OR is greater than 1, then the event is associated with a higher odds of generating a specific outcome. Conversely, if the OR is less than 1, then the event is associated with a lower odds of that outcome occurring. Based on the equation from above, the interpretation of an odds ratio can be denoted as the following: the odds of a success changes by exp(cB_1) times for every c-unit increase in x.

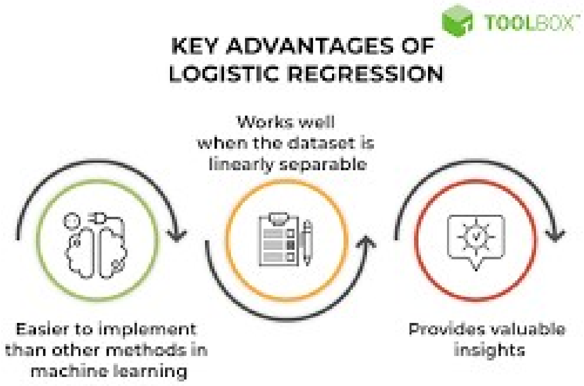
Types of logistic regression

There are three types of logistic regression models, which are defined based on categorical response.

- *Binary logistic regression:* In this approach, the response or dependent variable is dichotomous in nature—i.e. it has only two possible outcomes (e.g. 0 or 1). Some popular examples of its use include predicting if an e-mail is spam or not spam or if a tumor is malignant or not malignant. Within logistic regression, this is the most commonly used approach, and more generally, it is one of the most common classifiers for binary classification.

- *Multinomial logistic regression:* In this type of logistic regression model, the dependent variable has three or more possible outcomes; however, these values have no specified order. For example, movie studios want to predict what genre of film a moviegoer is likely to see to market films more effectively. A multinomial logistic regression model can help the studio to determine the strength of influence a person's age, gender, and dating status may have on the type of film that they prefer. The studio can then orient an advertising campaign of a specific movie toward a group of people likely to go see it.

- *Ordinal logistic regression:* This type of logistic regression model is leveraged when the response variable has three or more possible outcome, but in this case, these values do have a defined order. Examples of ordinal responses include grading scales from A to F or rating scales from 1 to 5.

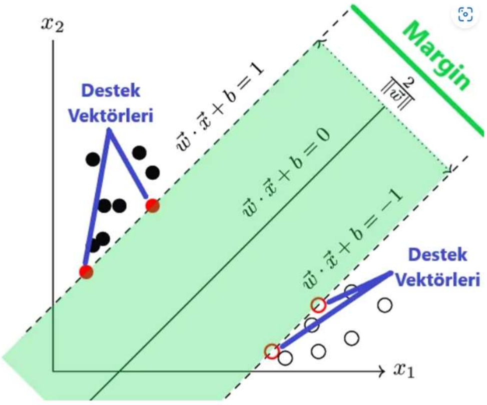

TOOLBOX
KEY ADVANTAGES OF LOGISTIC REGRESSION
Works well when the dataset is linearly separable
Easier to implement than other methods in machine learning
Provides valuable insights

Advantages and Disadvantages of Logistic Regression:

Advantages	Disadvantages
Logistic regression is easier to implement, interpret, and very efficient to train.	If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting.
It makes no assumptions about distributions of classes in feature space.	It constructs linear boundaries.
It can easily extend to multiple classes(multinomial regression) and a natural probabilistic view of class predictions.	The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables.
It not only provides a measure of how appropriate a predictor(coefficient size)is, but also its direction of association (positive or negative).	It can only be used to predict discrete functions. Hence, the dependent variable of Logistic Regression is bound to the discrete number set.
It is very fast at classifying unknown records.	Non-linear problems can’t be solved with logistic regression because it has a linear decision surface. Linearly separable data is rarely found in real-world scenarios.
Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.	Logistic Regression requires average or no multicollinearity between independent variables.
It can interpret model coefficients as indicators of feature importance.	It is tough to obtain complex relationships using logistic regression. More powerful and compact algorithms such as Neural Networks can easily outperform this algorithm.
Logistic regression is less inclined to over-fitting but it can overfit in high dimensional datasets.One may consider Regularization (L1 and L2) techniques to avoid over-fittingin these scenarios.	In Linear Regression independent and dependent variables are related linearly. But Logistic Regression needs that independent variables are linearly related to the log odds (log(p/(1-p)).

Support Vector Machine(SVM) Model:

Support Vector Machines are one of the supervised learning methods generally used in classification problems. Draws a line to separate points in the plane. It is aimed to provide the maximum distance for the points of both classes of this line. SVM is widely applied in various domains, including image classification, text classification, and bioinformatics.


x_2
Destek Vektörleri
$\vec{w}\cdot\vec{x}+b=1$
Margin
$\frac{2}{\|\vec{w}\|}$
$\vec{w}\cdot\vec{x}+b=0$
$\vec{w}\cdot\vec{x}+b=-1$
Destek Vektörleri
x_1

There are two different classes in the painting: blacks and whites. Our main purpose in classification problems is to decide in which class the future data will be placed. In order to make this classification, a line is drawn separating the two classes and the green region between ±1 of this line is called Margin. The wider the margin, the better the two or more classes are separated.

Why use support vector machines?

While creating decision boundaries, data belonging to two groups are selected at the edges of the boundaries. This serves as a vector of data signals and enables edge detection. This is where the name support vector machines comes from.

The SVM data set is divided into two according to the linear circulation.

Linear SVM:

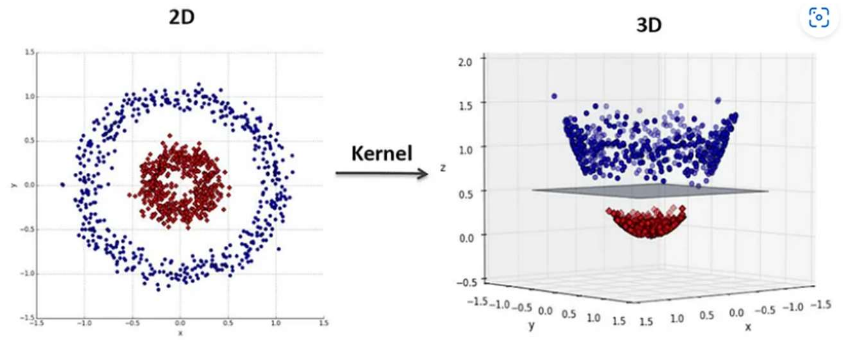
Linear SVM is used for linearly separable data sets. As can be seen from the image below, a straight line can effectively separate and classify two groups.

Nonlinear SVM:

Nonlinear SVM is used for data sets that cannot be separated by a straight line. Since a linear hyperplane cannot be used, structures called "kernel tricks" are used. In this way, a high classification rate is achieved. There are five types of kernel tricks: 'linear', 'poly', 'rbf', 'sigmoid' and 'precomputed', but the most used ones are 'poly' and 'rbf'.

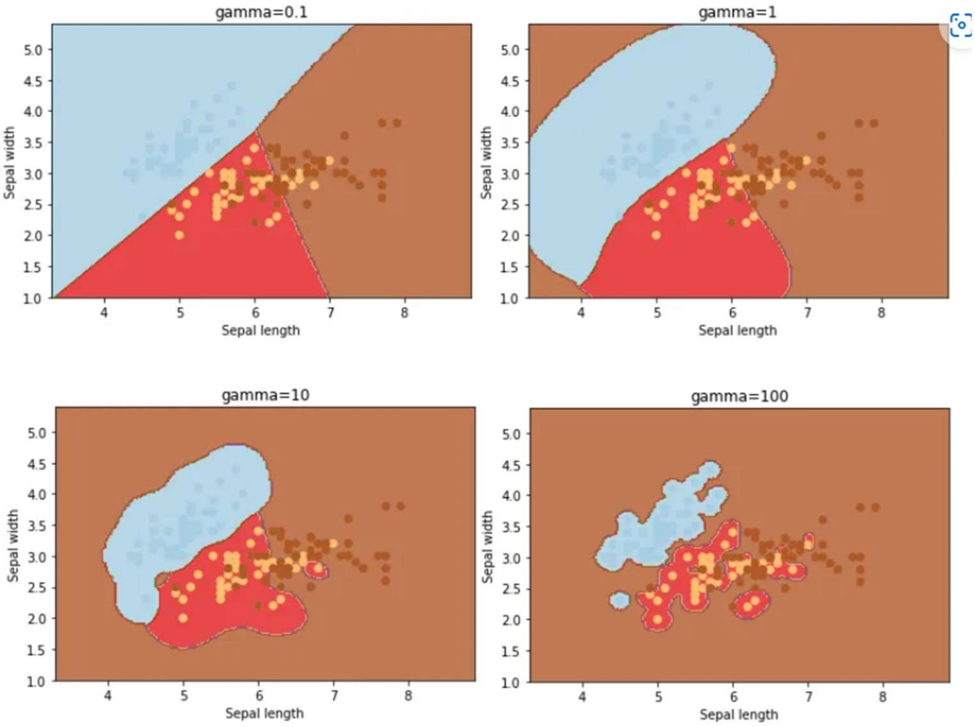
Poly:

It comes from polynomials and enables the learning of non-linear models by representing the similarity of the training data on polynomials.



RBF (Radial Basis Function):

RBF is one of the most widely used kernels due to its similarity to the Gaussian distribution. It calculates the similarity of two points in the data set according to their positions.

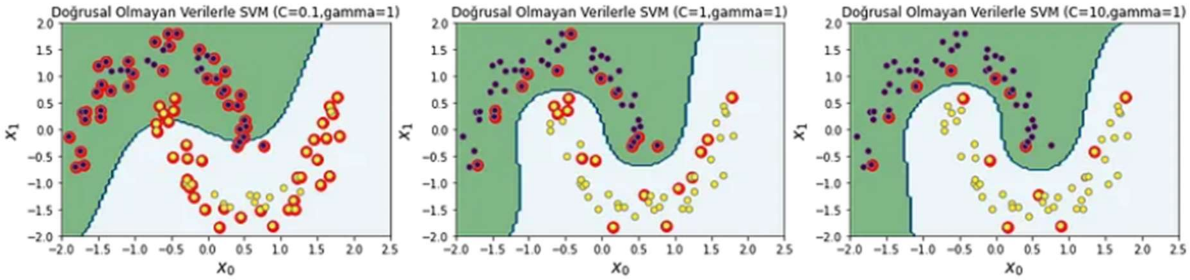


Hyperparameters:

- **'C' Hyperparameter:**

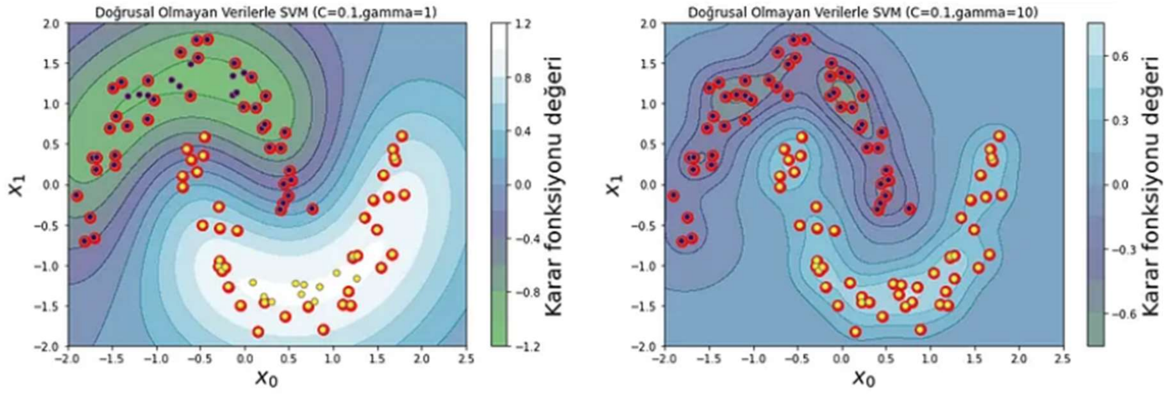
We can observe that the area expressed as the margin width is the space between the support vectors of both groups. The narrower this area, the sharper our classification can be. This is referred to as 'hard margin.' However, on the one hand, a narrow-margin classification can lead to errors depending on the distribution of the dataset or whether it is linear or not. Here, by increasing this margin width, we can apply a broader margin called 'soft margin.' In this way, some training data may remain within the margin even though it is not a support vector, but this will result in more accurate classifications during the process. We can control this margin width with the C parameter. The larger the C, the narrower the margin; the smaller the C, the larger the margin will be.

When classifying non-linear data structures with minimum error, we should pay attention to ensuring that the 'C' parameter is small enough to be a 'soft margin' and large enough not to overfit. Finding the optimal balance is crucial.



- **'gamma' Hyperparameters:**

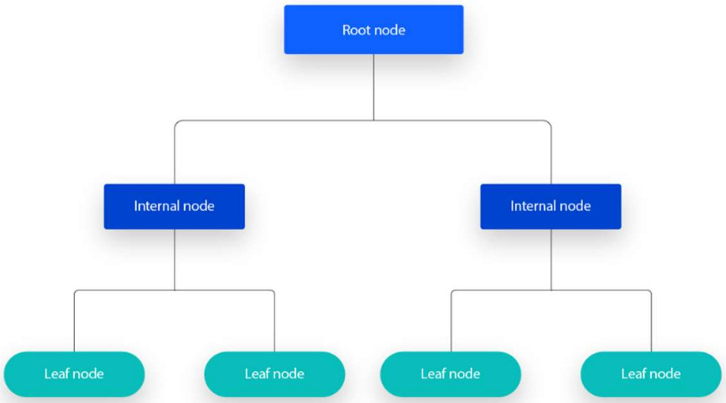
We will need the 'gamma' parameter more when modeling non-linear data structures. When working with different kernel types, we can control the width of the normal distribution used in classification with the 'gamma' parameter. If we examine the visual below, we can see the difference between 'gamma = 1' and 'gamma = 10'. The smaller the 'gamma,' the wider the distribution; the larger the 'gamma,' the narrower the distribution. In other words, the 'gamma' value should be large enough for maximum classification quality but small enough to avoid overfitting. In case of overfitting, we should decrease it, and in the case of underfitting, we should increase it.



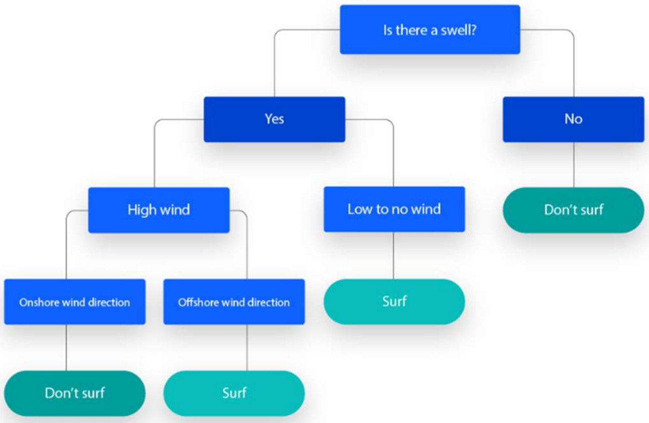
DECISION TREE

1) What Is Decision Tree

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.



A decision tree commences with a root node devoid of incoming branches. Subsequently, outgoing branches stemming from the root node extend into internal nodes, also referred to as decision nodes. Both root and internal nodes evaluate available features to partition the data into homogenous subsets, characterized by leaf nodes or terminal nodes. These leaf nodes encompass all feasible outcomes present within the dataset.

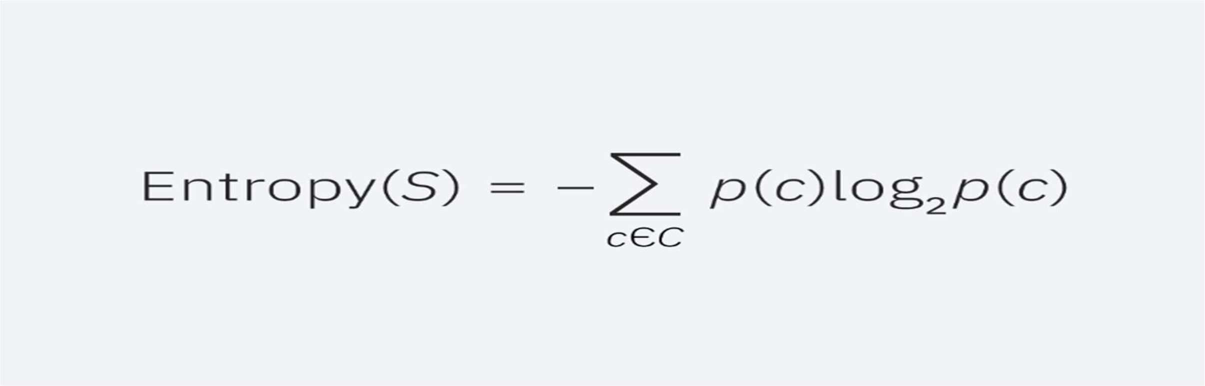


2)How to choose the best attribute at each node?

Entropy and Information Gain.

i)What is Entropy in Decision Tree?

In decision trees, entropy is a measure of impurity used to evaluate the homogeneity of a dataset. It helps determine the best split for building an informative decision tree model.

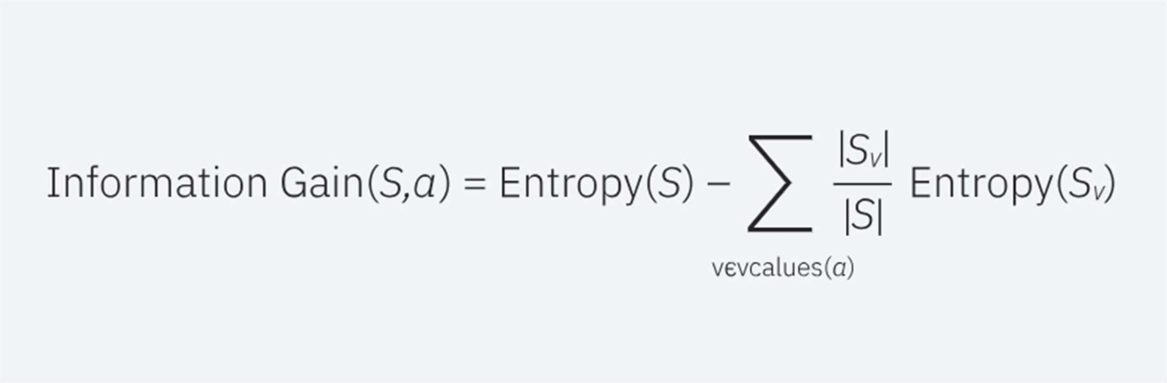


$$\mathrm{Entropy}(S) = -\sum_{c \in C} p(c)\log_2 p(c)$$

S represents the data set that entropy is calculated
c represents the classes in set, S
p(c) represents the proportion of data points that belong to class c to the number of total data points in set, S

ii)What is Information Gain in Decision Tree?

Information gain is a measure used to determine which feature should be used to split the data at each internal node of the decision tree. It is calculated using entropy.



$$\mathrm{Information\ Gain}(S,a) = \mathrm{Entropy}(S) - \sum_{v \in vcalues(a)} \frac{|S_v|}{|S|}\,\mathrm{Entropy}(S_v)$$

a represents a specific attribute or class label
Entropy(S) is the entropy of dataset, S
|Sv|/ |S| represents the proportion of the values in S_v to the number of values in dataset, S
Entropy(S_v) is the entropy of dataset, S_v

3)Advantages and disadvantages of Decision Trees

Advantages

- Easy to interpret: The Boolean logic and visual representations of decision trees make them easier to understand and consume. The hierarchical nature of a decision tree also makes it easy to see which attributes are most important, which isn't always clear with other algorithms, like neural networks .

- Little to no data preparation required: Decision trees have a number of characteristics, which make it more flexible than other classifiers. It can handle various data types—i.e. discrete or continuous values, and continuous values can be converted into categorical values through the use of thresholds. Additionally, it can also handle values with missing values, which can be problematic for other classifiers, like Naïve Bayes.

- More flexible: Decision trees can be leveraged for both classification and regression tasks, making it more flexible than some other algorithms. It's also insensitive to underlying relationships between attributes; this means that if two variables are highly correlated, the algorithm will only choose one of the features to split on.

Disadvantages

- Prone to overfitting: Complex decision trees tend to overfit and do not generalize well to new data. This scenario can be avoided through the processes of pre-pruning or post-pruning. Pre-pruning halts tree growth when there is insufficient data while post-pruning removes subtrees with inadequate data after tree construction.

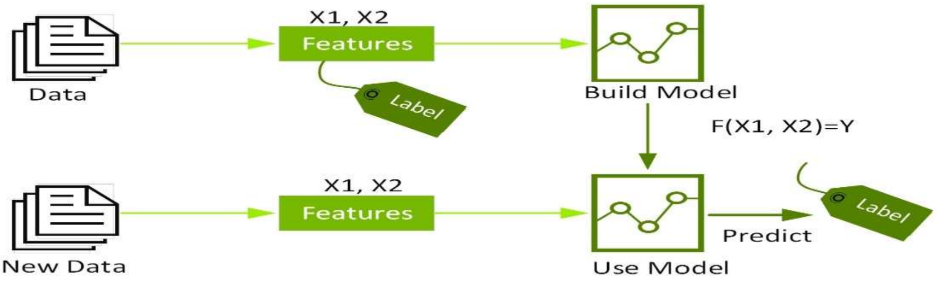
- High variance estimators: Small variations within data can produce a very different decision tree. Bagging, or the averaging of estimates, can be a method of reducing variance of decision trees. However, this approach is limited as it can lead to highly correlated predictors.

- More costly: Given that decision trees take a greedy search approach during construction, they can be more expensive to train compared to other algorithms.

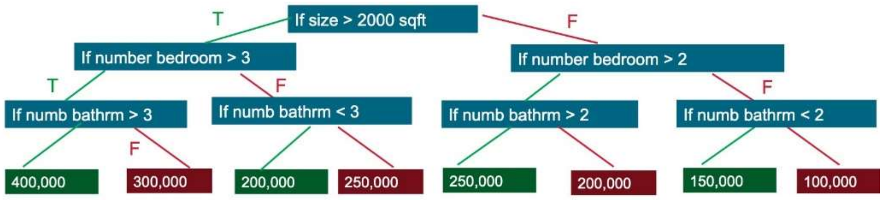
XGBOOST ALGORITHM

1)What is XGBoost Algorithm?

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distribute gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. It’s vital to an understanding of XGBoost to first grasp the machine learning concepts and algorithms that XGBoost builds upon: supervised machine learning, decision trees, ensemble learning, and gradient boosting. Supervised machine learning uses algorithms to train a model to find patterns in a dataset with labels and features and then uses the trained model to predict the labels on a new dataset’s features.

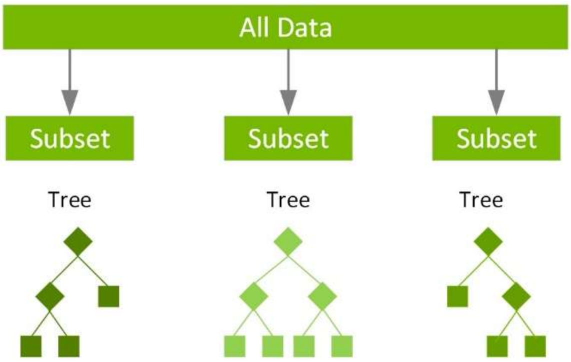


Decision trees create a model that predicts the label by evaluating a tree of if-then-else true/false feature questions, and estimating the minimum number of questions needed to assess the probability of making a correct decision. Decision trees can be used for classification to predict a category, or regression to predict a continuous numeric value. In the simple example below, a decision tree is used to estimate a house price (the label) based on the size and number of bedrooms (the features).



A Gradient Boosting Decision Trees (GBDT) is a decision tree ensemble learning algorithm similar to random forest, for classification and regression.

Ensemble learning algorithms combine multiple machine learning algorithms to obtain a better model.
Both random forest and GBDT build a model consisting of multiple decision trees. The difference is in how the trees are built and combined.



The term “gradient boosting” comes from the idea of “boosting” or improving a single weak model by combining it with a number of other weak models in order to generate a collectively strong model. Gradient boosting is an extension of boosting where the process of additively generating weak models is formalized as a gradient descent algorithm over an objective function. Gradient boosting sets targeted outcomes for the next model in an effort to minimize errors. Targeted outcomes for each case are based on the gradient of the error (hence the name gradient boosting) with respect to the prediction.

2)Why XGBoost?

XGBoost is used for these two reasons: execution speed and model performance. Execution speed is crucial because it's essential to working with large datasets. When you use XGBoost, there are no restrictions on the size of your dataset, so you can work with datasets that are larger than what would be possible with other algorithms. Model performance is also essential because it allows you to create models that can perform better than other models. XGBoost has been compared to different algorithms such as random forest (RF), gradient boosting machines (GBM), and gradient boosting decision trees (GBDT). These comparisons show that XGBoost outperforms these other algorithms in execution speed and model performance.

3)Advantages and Disadvantages of XGBoost

Advantages:

1. **Regularization**: XGBoost has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from overfitting. That is why, XGBoost is also called regularized form of GBM (Gradient Boosting Machine).

2. **Handling Missing Values**: XGBoost has an in-built capability to handle missing values. When XGBoost encounters a missing value at a node, it tries both the left and right hand split and learns the way leading to higher loss for each node. It then does the same when working on the testing data.
3. **Parallel Processing**: XGBoost utilizes the power of parallel processing and that is why it is much faster than GBM. It uses multiple CPU cores to execute the model.

Disadvantages:

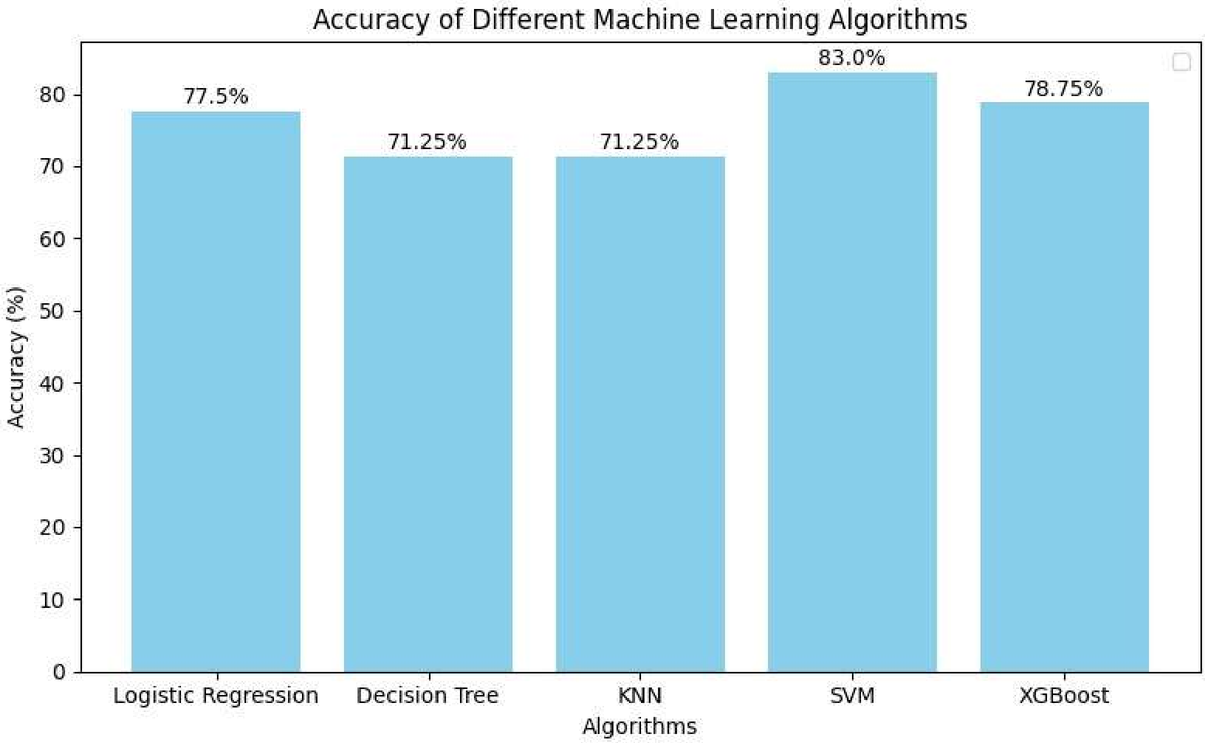
1. **Overfitting**: Overfiting is likely to occur in xgboost if xgboost parameters are not tuned properly.
2. **Training time**: Training time is pretty high for larger dataset, if you compare against catboost/lightgbm.

CONCLUSION

Based on the provided data, the accuracy rates obtained from different machine learning algorithms are as follows:

- Logistic Regression: 77.50%
- Decision Tree: 71.25%
- K-Nearest Neighbors (KNN): 71.25%
- Support Vector Machine (SVM): 83.00%
- XGBoost: 78.75%

This result shows that the SVM algorithm achieved the highest accuracy rate in our data set. Among other algorithms, XGBoost ranks second with the highest accuracy, while logistic regression ranks third. Decision tree and KNN algorithms exhibit relatively lower accuracy performance among the models considered. This graphical analysis reveals that SVM outperforms other models in terms of prediction accuracy for this dataset.



SOURCES

DATASET:

[1]https://www.archive.ics.uci.edu/dataset/862/turkish+music+emotion

[2] Bilal Er, M., & Aydilek, I. B. (2019). Music emotion recognition by using chroma spectrogram and deep visual features. Journal of Computational Intelligent Systems, 12(2), 1622â€"1634. International Journal of Computational Intelligence Systems, , DOI: https://doi.org/10.2991/ijcis.d.191216.001

KNN:

[1]https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[2]https://arslanev.medium.com/makine-%C3%B6%C4%9Frenmesi-knn-k-nearest-neighbors-algoritmas%C4%B1-bdfb688d7c5f

[3]https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/

[4]https://medium.com/@anuuz.soni/advantages-and-disadvantages-of-knn-ee06599b9336

[5]https://towardsdatascience.com/k-nearest-neighbours-explained-7c49853633b6

[6]https://builtin.com/machine-learning/nearest-neighbor-algorithm

[7]https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn

LOGISTIC REGRESSION:

[1]https://www.sciencedirect.com/topics/computer-science/logistic-regression#:~:text=Logistic%20regression%20is%20a%20process,%2Fno%2C%20and%20so%20on.

[2]https://en.wikipedia.org/wiki/Logistic_regression

[3]https://www.ibm.com/topics/logistic-regression

[4]https://aws.amazon.com/tr/what-is/logistic-regression/

SVM:

[1]https://scikitlearn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC

[2]https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

[3]https://tr.qaz.wiki/wiki/Support_vector_machine

DECISION TREE:

[1]https://www.ibm.com/spss

[2]https://medium.com/

[3]https://www.analyticsvidhya.com/blog/

XGB:

[1]https://www.nvidia.com/en-us/glossary/data-science/xgboost/

[2]https://www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article#:~:text=XGBoost%20is%20a%20classification%20algorithm,that%20you%20want%20to%20classify.

[3]https://vatsalparsaniya.com/ML_Knowledge/XGBoost/Readme.html