“Optimization problems for machine learning: A survey”

By: Mohammad Heydari
SID:810197494
Dr. Hadi Amiri

“Abstract”

- This paper surveys the machine learning literature and presents in an optimization framework several commonly used machine learning approaches. Particularly, mathematical optimization models are presented for regression, classification, clustering, deep learning, and adversarial learning, as well as new emerging applications in machine teaching, empirical model learning, and Bayesian network structure learning. Such models can benefit from the advancement of numerical optimization techniques which have already played a distinctive role in several machine learning settings. The strengths and the shortcomings of these models are discussed and potential research directions and open problems are highlighted.

1. Regression models

- A regression model provides **a function that describes the relationship between one or more independent variables and a response, dependent, or target variable**. ... A regression analysis is the basis for many types of prediction and for determining the effects on target variables.
- The most commonly used loss function for regression is the least squared estimate, where fitting a regression model reduces to minimizing the residual sum of squares (RSS) between the labels and the predicted outputs, such as

$$RSS(\beta) = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 .$$

2. Classification

- The task of classifying data is to decide the class membership of an unlabeled data item x based on the training dataset (X, y) where each xi has a known class membership yi.

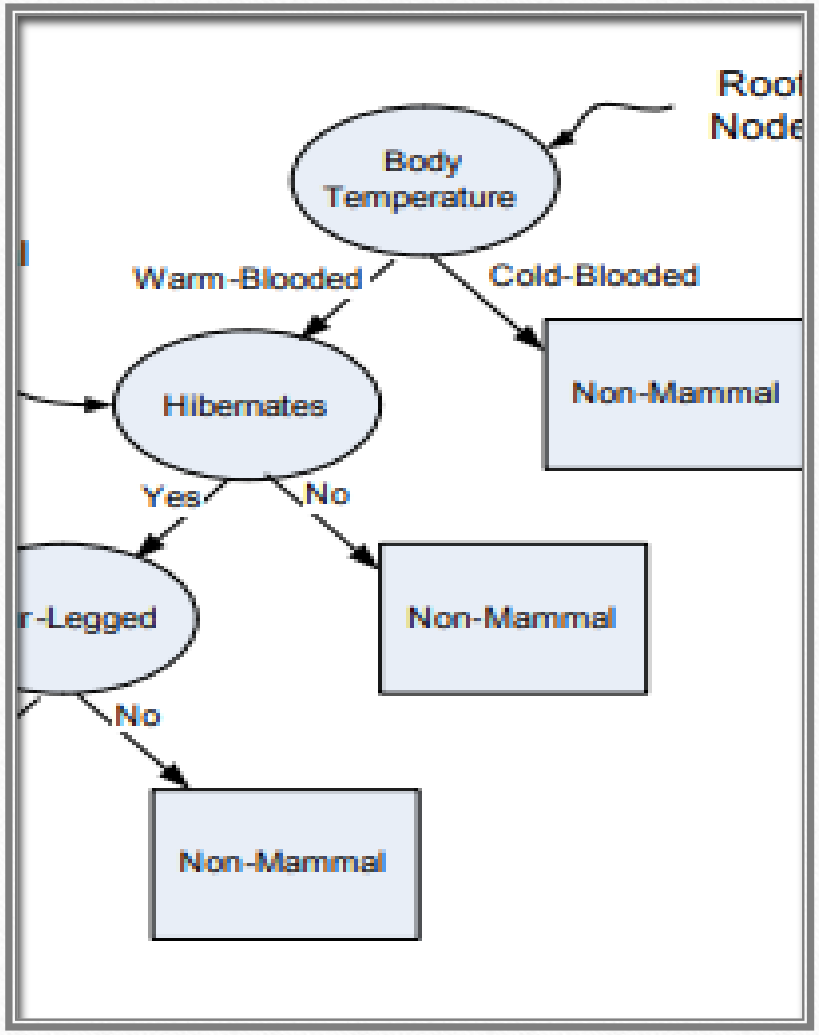
- A recent comparison of machine learning techniques for binary classification is found in Baumann, Hochbaum, and Yang (2019). This section reviews the common binary and multiclass classification approaches that include logistic regression, linear discriminant analysis, decision trees, and support vector machines.

3. Decision trees

What is a Decision Tree?

- Choose the attribute that minimize the Disorder in the sub-tree rooted at a given node.

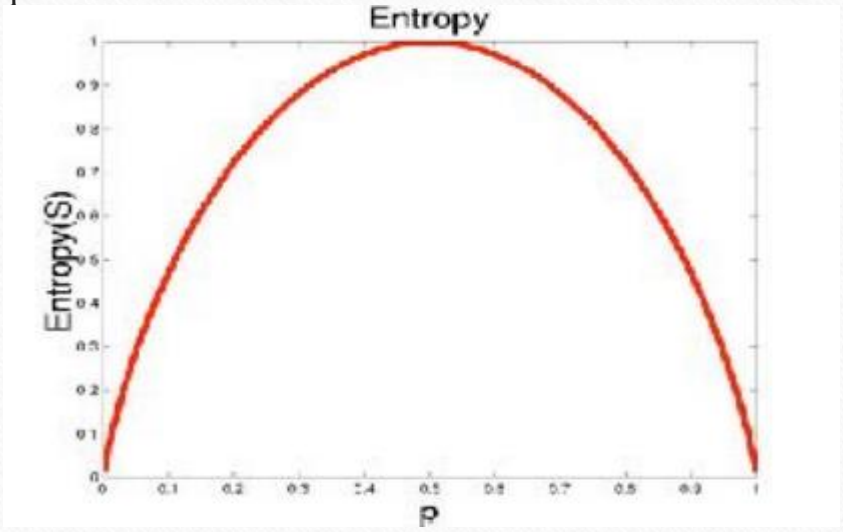
- Disorder and Information are related as follows: the more disorderly a set, the more information is required to correctly guess an element of that set.

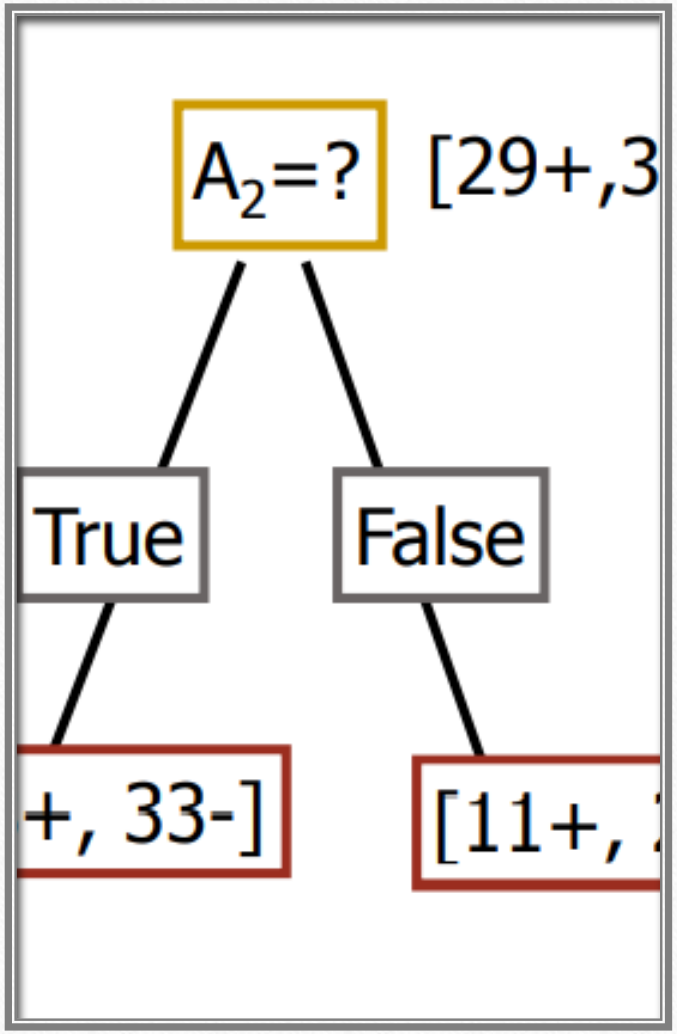


“Entropy”

S is a sample of training examples

- p+ is the proportion of positive examples
- p- is the proportion of negative examples
- Entropy measures the impurity of S Entropy(S) = -p+ log2 p+ - p- log2 p-





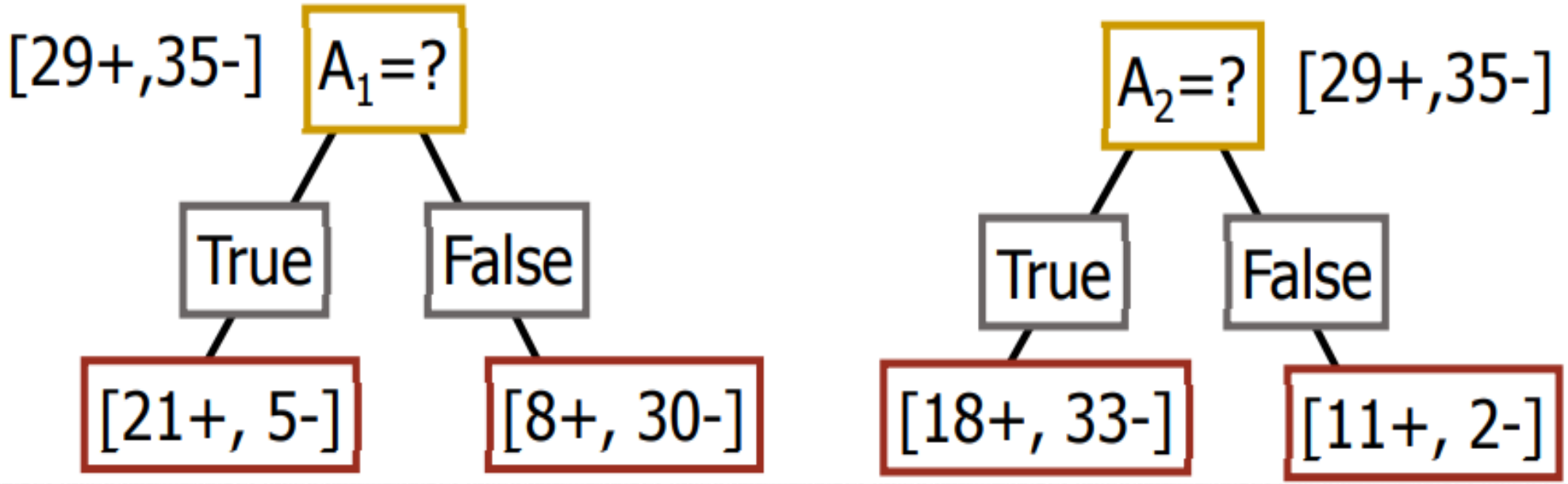
Information Gain

- In decision tree learning, Information gain ratio is a ratio of information gain to the intrinsic information. It was proposed by Ross Quinlan, to reduce a bias towards multi-valued attributes by taking the number and size of branches into account when choosing an attribute.

Gain(S,A): expected reduction in entropy due to sorting S on attribute A

$$Gain(S,A) = Entropy(S) - \sum_{v \in values(A)} |S_v|/|S| \; Entropy(S_v)$$

$$Entropy([29+,35-]) = -29/64 \log_2 29/64 - 35/64 \log_2 35/64$$
$$= 0.99$$



$$\min \frac{1}{\hat{L}} \sum_{t \in T_L} L_t + \alpha \sum_{t \in T_B} d_t \tag{22}$$

$$\text{s.t.} \quad L_t \geq N_t - N_{kt} - n(1 - c_{kt}), \quad \forall k = 1, \ldots, K, \ t \in T_L, \tag{23}$$

$$0 \leq L_t \leq N_t - N_{kt} + n c_{kt} \quad \forall k = 1, \ldots, K, \ t \in T_L, \tag{24}$$

$$N_{kt} = \frac{1}{2} \sum_{i=1}^{n} (1 + Y_{ik}) z_{it}, \quad \forall k = 1, \ldots, K, \ t \in T_L, \tag{25}$$

$$N_t = \sum_{i=1}^{n} z_{it} \quad \forall t \in T_L, \tag{26}$$

$$\sum_{k=1}^{K} c_{kt} = l_t \quad \forall t \in T_L, \tag{27}$$

$$\sum_{t \in T_L} z_{it} = 1 \quad \forall i = 1, \ldots, n, \tag{28}$$

$$z_{it} \leq l_t \quad \forall i = 1, \ldots, n, \ t \in T_L, \tag{29}$$

$$\sum_{i=1}^{n} z_{it} \geq N_{\min} l_t \quad \forall t \in T_L, \tag{30}$$

$$a_m^\top (x_i + \epsilon) \leq b_m + (1 + \epsilon_{\max})(1 - z_{it}) \quad \forall i = 1, \ldots, n, \ t \in T_L, \ m \in A_L(t), \tag{31}$$

$$a_m^\top x_i \geq b_m - (1 - z_{it}) \quad \forall i = 1, \ldots, n, \ t \in T_L, \ \forall m \in A_R(t), \tag{32}$$

$$\sum_{j=1}^{p} a_{jt} = d_t \quad \forall t \in T_B, \tag{33}$$

$$0 \leq b_t \leq d_t \quad \forall t \in T_B, \tag{34}$$

$$d_t \leq d_{p(t)} \quad \forall t \in T_B \setminus \{1\}, \tag{35}$$

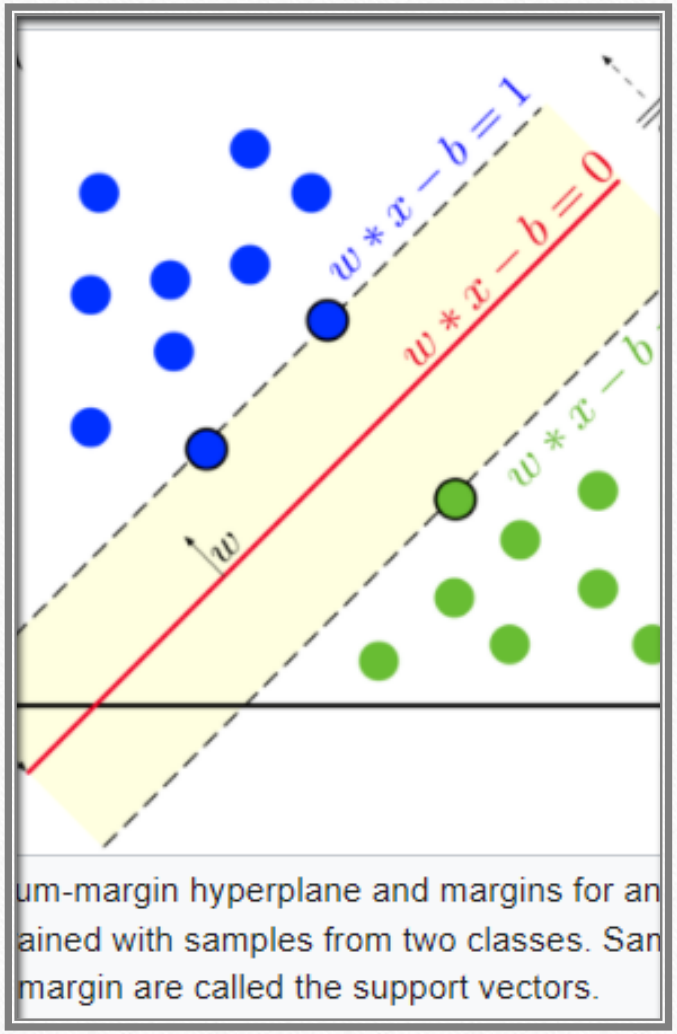
$$z_{it}, \ l_t \in \{0, 1\} \quad \forall i = 1, \ldots, n, \ \forall t \in T_L, \tag{36}$$

$$c_{kt} \in \{0, 1\} \quad \forall k = 1, \ldots, K, \ t \in T_L, \tag{37}$$

$$a_{jt}, \ d_t \in \{0, 1\} \quad \forall j = 1, \ldots, p, \ t \in T_B. \tag{38}$$

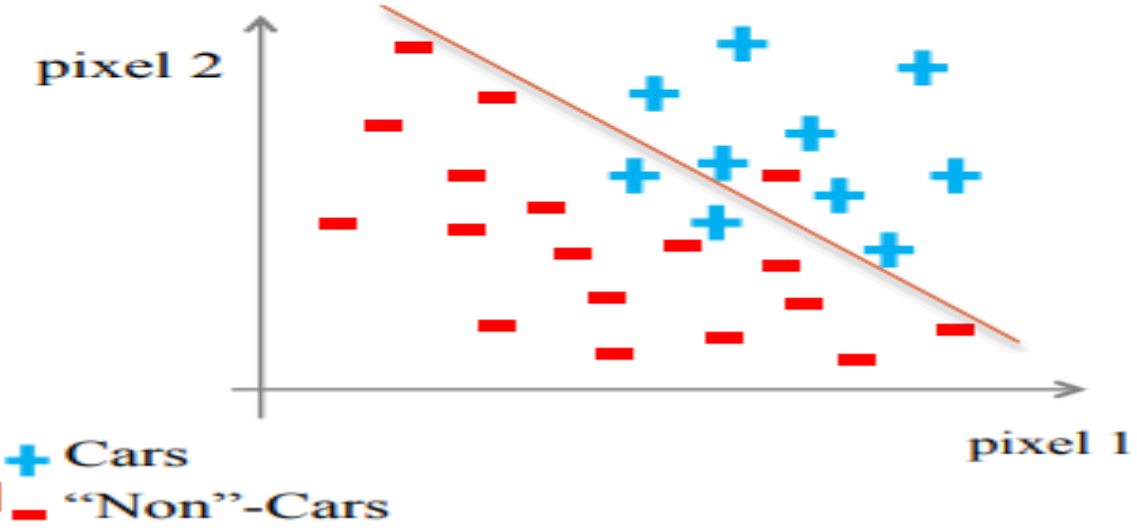
4. Support vector regression

Although as discussed earlier, SVM has been introduced for binary classification, its extension to regression, i.e., support vector regression, has received significant interest in the literature (Smola & Schölkopf, 2004). The core idea of support vector regression is to find a linear function $f(x) = wx + \gamma$ that can approximate with a tolerance a training set (X, y) where $y \in R$ (Vapnik, 2013). Such a linear function may however not exist, and thus slack variables $\xi + i \geq 0$ and $\xi - i \geq 0$ denoting positive and negative deviations from the desired tolerance are introduced and minimized similar to the soft-margin SVM. The corresponding optimization problem is



um-margin hyperplane and margins for an
ained with samples from two classes. San
margin are called the support vectors.

Learning Algorithm



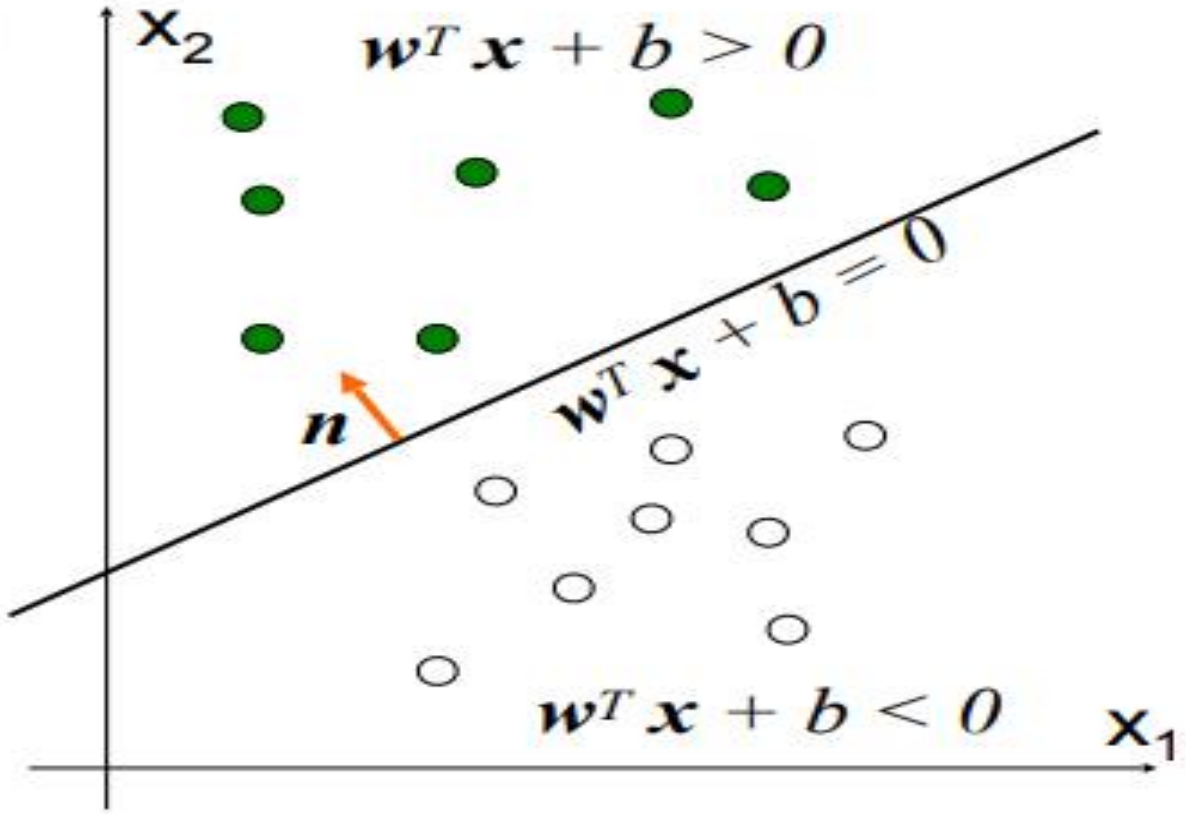
50 x 50 pixel images→ 2500 pixels

(7500 if RGB)

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$

Linear Classifier

- Discriminant function is linear
- A hyper-plane in feature space



Support Vector Machine

- In practice, the data are not separable.
- The minimization problem can still be solved
- In practice, the parameter C is found using crossvalidation

✓ $L(\mathbf{w},b,c)=0.5||\mathbf{w}||^2+C\sum\max(0,1-y_i(\mathbf{w}^T\mathbf{x}_i + b))$

Maximize the margin

- In order to maximize the margin, we need to minimize ||w||2.

With the condition that there are no datapoints between H1 and H2:

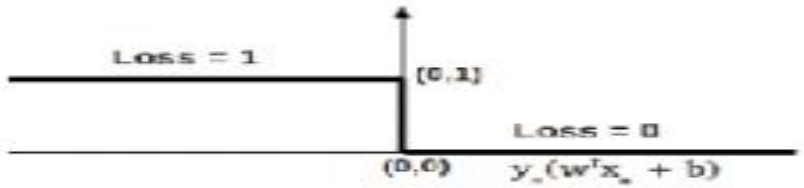
✓ wTxi + b≥ +1 when yi =+1 ✓ wTxi + b ≤ -1 when yi =-1

• Lagrangian method: minimize L(w,b,C) ✓ L(w,b,c)=0.5||w||2+C∑max(0,1-yi(wTxi + b)) ✓ For sufficiently large C

Loss Functions

Zero-One Loss

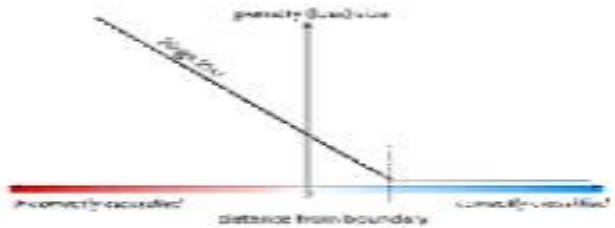
$$l(\hat{y}, y) = \mathbf{1}[\text{sign}(\hat{y}) = y]$$



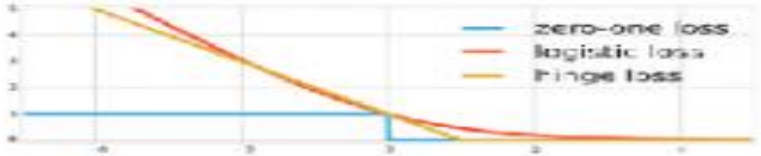
SVM Loss Function

Hinge Loss

$$\ell(\hat{y}, y) = \max(0, 1 - y \cdot \hat{y})$$



Logistic Loss $l(\hat{y}, y) = \log(1 + \exp(1 - y.\hat{y}))$



- ✓ $L(\mathbf{w},b,c)=0.5\|\mathbf{w}\|^2+C\sum\max(0,1-y_i(\mathbf{w}^T\mathbf{x}_i + b))$

$$\min \quad \|w\|_2^2 + C \sum_{j=1}^{r-1} \left(\sum_{k=1}^{j} \sum_{i=1}^{n_k} \xi_{i,kj}^{+} + \sum_{k=j+1}^{r} \sum_{i=1}^{n_k} \xi_{i,kj}^{-} \right)$$

$$\text{s.t.} \quad w^\top x_{i,k} - \beta_j \leq -1 + \xi_{i,kj}^{+}$$

$$\forall k = 1, \ldots, j, \;\; j = 1, \ldots, r-1, \;\; i = 1, \ldots, n_k,$$

$$w^\top x_{i,k} - \beta_j \geq 1 - \xi_{i,kj}^{-}$$

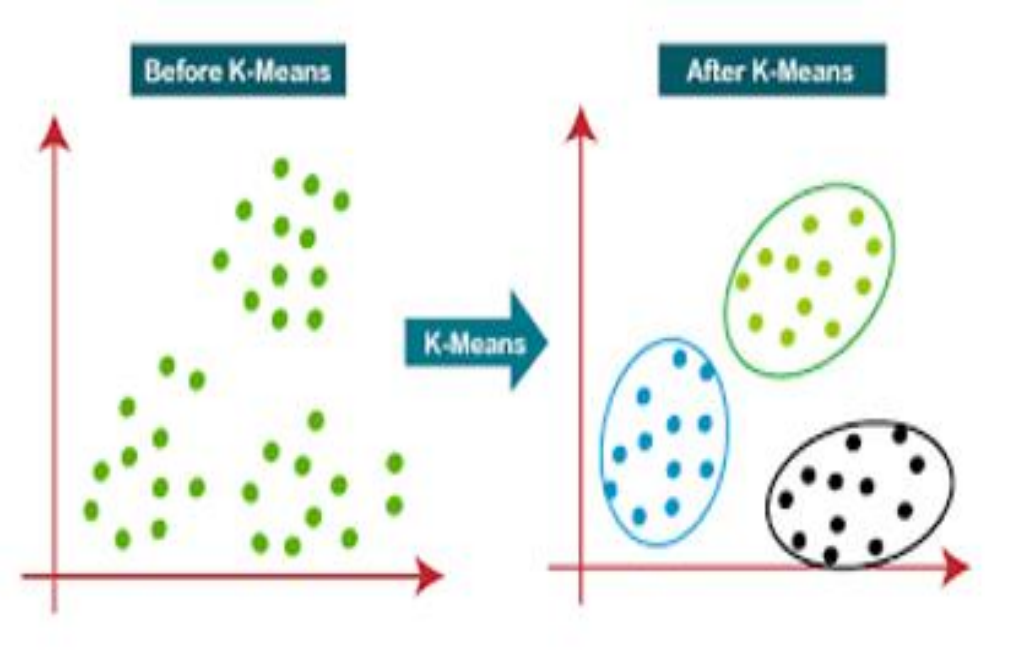
$$\forall k = j+1, \ldots, r, \;\; j = 1, \ldots, r-1, \;\; i = 1, \ldots, n_k,$$

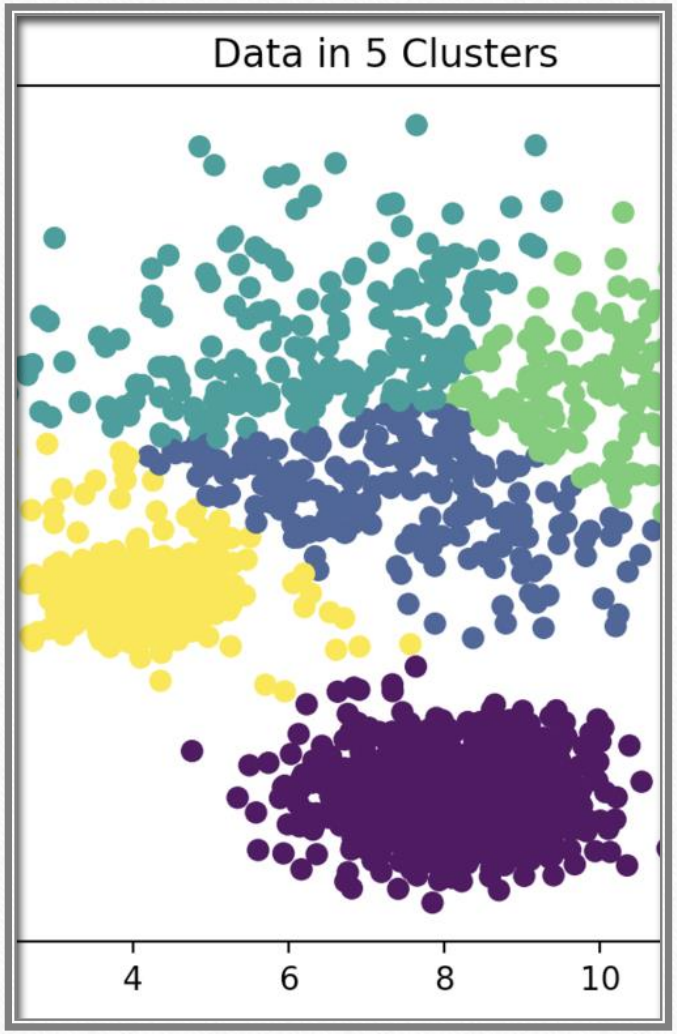
$$w \in \mathbb{R}^p, \;\; \beta_j \in \mathbb{R} \quad \forall j = 1, \ldots, r-1,$$

$$\xi_{i,kj}^{+} \geq 0 \quad \forall k = 1, \ldots, j, \;\; j = 1, \ldots, r-1, \;\; i = 1, \ldots, n_k,$$

$$\xi_{i,kj}^{-} \geq 0 \quad \forall k = j+1, \ldots, r, \;\; j = 1, \ldots, r-1, \;\; i = 1, \ldots, n_k.$$

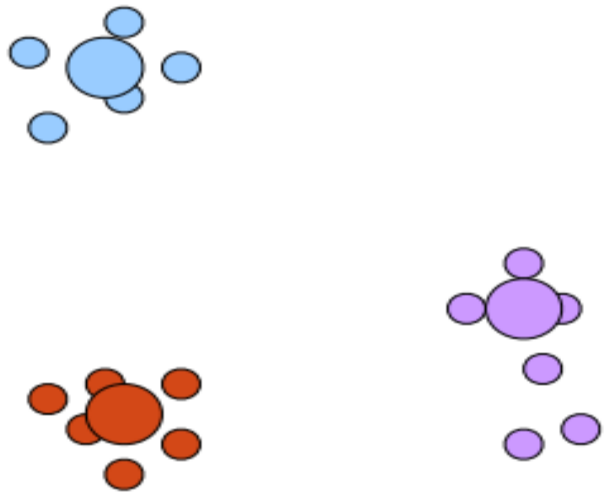
5. Clustering(K-means)





K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate Clusters
- Repeat…



K-means Optimization Objective

$C(i)$ = index of cluster (1, 2, …, K) to which example $\mathbf{x}_i$ is currently assigned

$\mathbf{m}_k$ = cluster centroid k ($\mathbf{m}_k \in \mathbb{R}^n$)

$\mathbf{m}_{C(i)}$ = cluster centroid of cluster to which example $\mathbf{x}_i$ has been assigned

Optimization objective:

$$J(C(1),\ldots,C(N),m_1,\ldots,m_K) = \frac{1}{N}\sum_{i=1}^{N}\|x_i - m_{C(i)}\|^2$$

$$\min_{\substack{C(1),\ldots,C(N)\\ m_1,\ldots,m_K}} J(C(1),\ldots,C(N),m_1,\ldots,m_K)$$

Within and Between Cluster Criteria

- Within Cluster and Between Cluster Scatter:

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k}^{N} d(x_i, x_j)^2$$

$$B(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)\neq k}^{N} d(x_i, x_j)^2$$

- If d is square Euclidean distance, then

$$W(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} \|x_i - m_k\|^2$$

$$m_k = \frac{1}{N_k} \sum_{C(i)=k} x_i$$

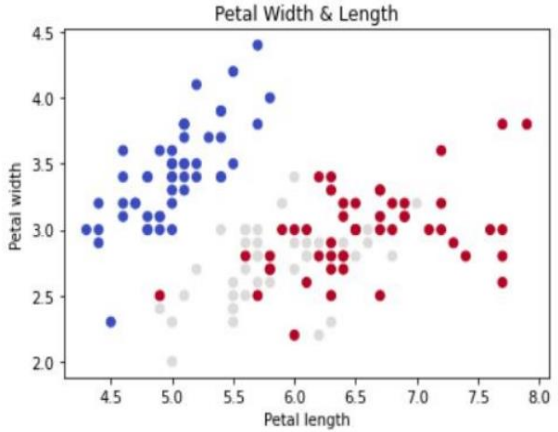
Bonus Part: (Simulations Algorithm)

- I have implemented some of the most common supervised and un-supervised learning algorithm in python and further you can find the results of some of them:
- 1)SVM
- 2) Decision Tree
- 3) K-means Clustering

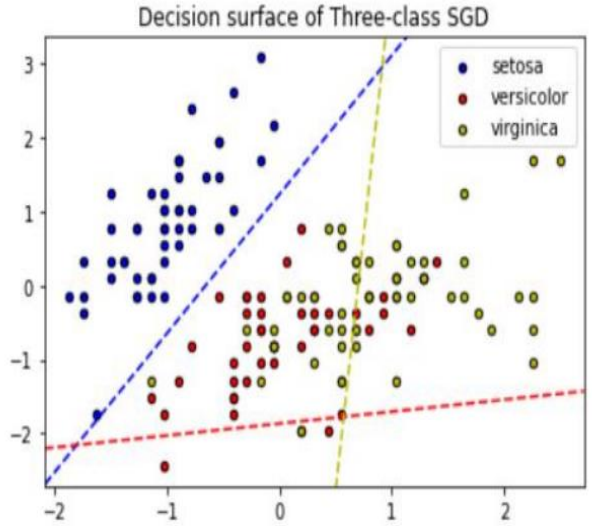
SVM(with libraries)

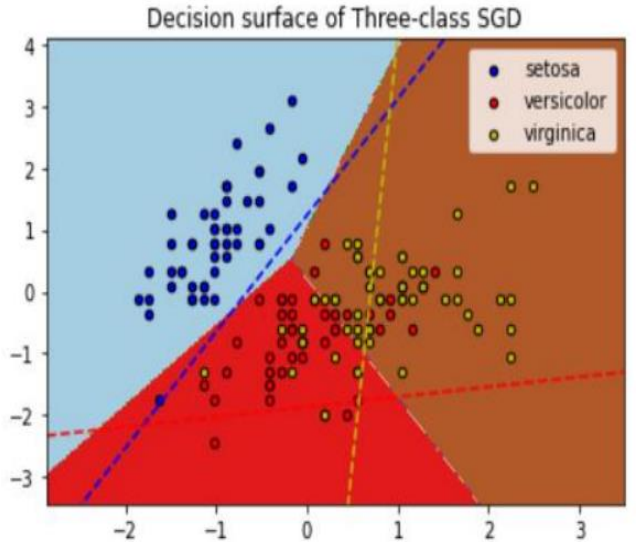
Implementing has been done in python and results have provided below:

Before classification:



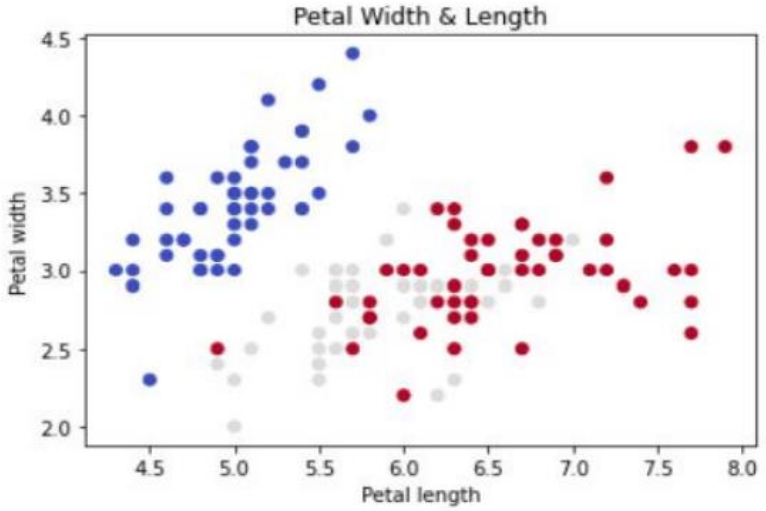
After classification:



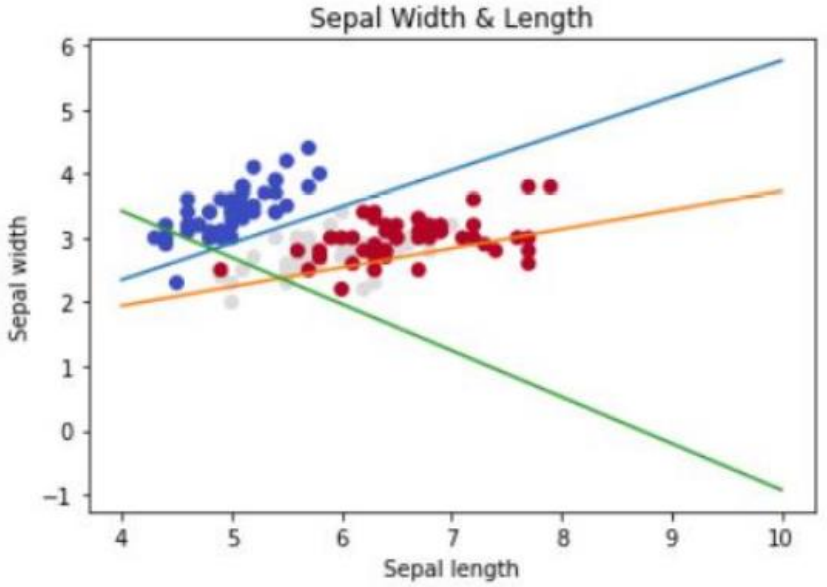


SVM(without libraries)

Before classification:



After classification:



Decision Tree(without libraries)

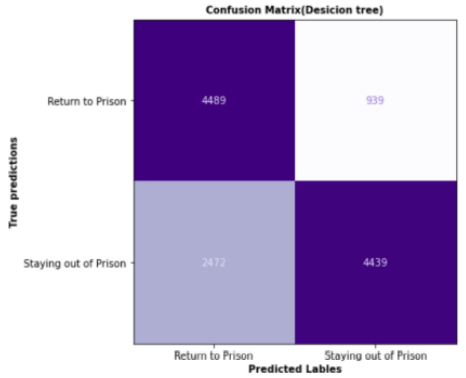
	Fiscal Year Released	Recidivism Reporting Year	Race - Ethnicity	Age At Release	Convicting Offense Classification	Convicting Offense Type	Convicting Offense Subtype	Main Supervising District	Release Type	Part of Target Population	Recidivism - Return to Prison numeric
0	2010	2013	White	<45	D Felony	Violent	Other	3JD	Parole	Yes	1
1	2010	2013	White	>45	D Felony	Other	Other	3JD	Parole	Yes	1
2	2010	2013	White	<45	D Felony	Other	Other	5JD	Parole	Yes	1
3	2010	2013	White	>45	Other Felony	Drug	Trafficking	3JD	Parole	Yes	1
4	2010	2013	Black	<45	D Felony	Drug	Trafficking	3JD	Parole	Yes	1
...	...	...	...	...	...	...	...	...	...	...	...
15419	2015	2018	White	<45	Other Felony	Violent	Other	3JD	Discharged End of Sentence	Yes	0
15420	2015	2018	White	<45	D Felony	Other	Other	5JD	Discharged End of Sentence	No	0
15421	2015	2018	Black	<45	Other Felony	Violent	Other	3JD	Discharged End of Sentence	Yes	0
15422	2015	2018	White	<45	D Felony	Drug	Other	5JD	Parole	No	0
15423	2015	2018	White	<45	Other Felony	Violent	Other	3JD	Parole	No	0

15424 rows × 11 columns

Figure 5: Dataset

```
Accuracy of Decision Tree is: % 72.36
```

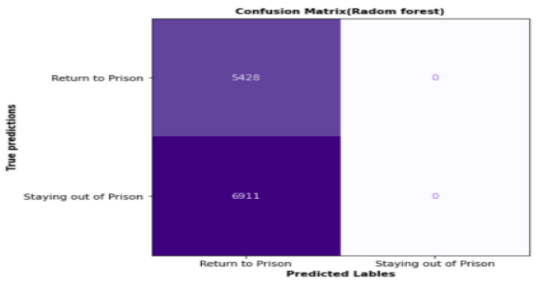
Confusion Matrix:



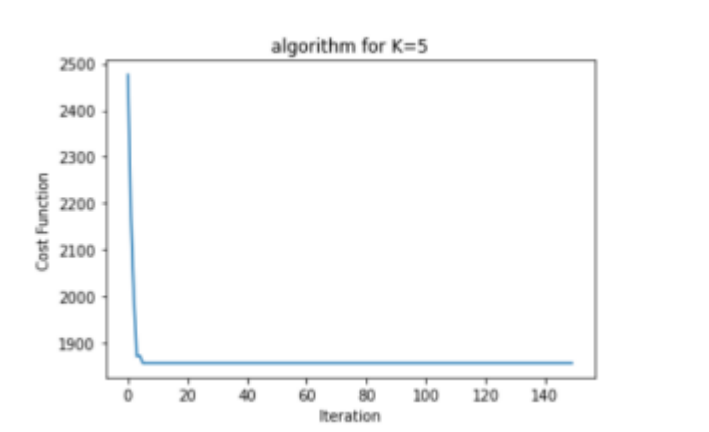
Results:

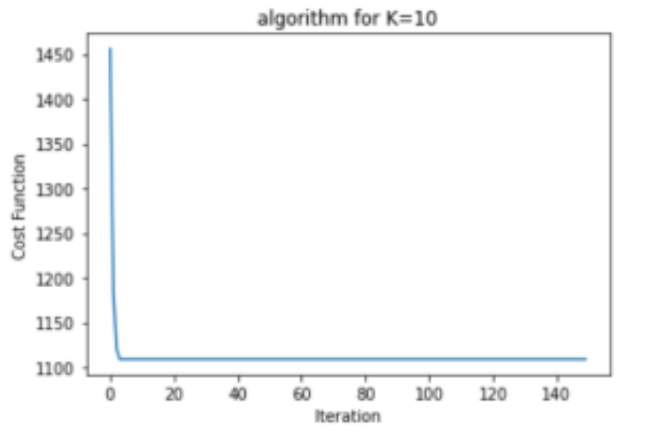
```
Accuracy of Random Forest with k =  3 is: % 43.99
```

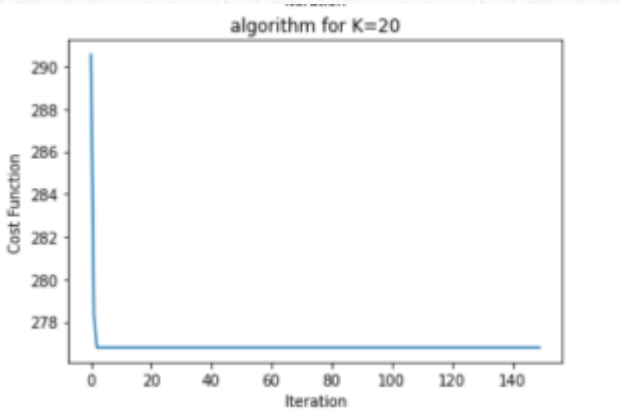
Confusion Matrix:



K-means Clustering(without libraries)







As we can see the algorithm convergence speed is very high but it's reasonable because our dataset is about length and width of plants and the density of these data as of our first observation is so high So despite we have used a random-based approach for initializing the first centroids but cause of mentioned statement the convergence speed is very high and as you can see under 20 iterations we meet the convergence!