Machine Learning Assignment 4

Team Members:

1. **Amir Youssef**
2. **Mohamed Elesawy**
3. **Abdulrahman Ahmed**

Part 1: Calculations

Weather (F1)	Temperature (F2)	Humidity (F3)	Wind (F4)	Hiking (Labels)
Cloudy	Cool	Normal	Weak	No
Sunny	Hot	High	Weak	Yes
Rainy	Mild	Normal	Strong	Yes
Cloudy	Mild	High	Strong	No
Sunny	Mild	High	Strong	No
Rainy	Cool	Normal	Strong	No
Cloudy	Mild	High	Weak	Yes
Sunny	Hot	High	Strong	No
Rainy	Cool	Normal	Weak	No
Sunny	Hot	High	Strong	No

a) Using Gini Index:

$$P\ (\text{Hiking} = \text{No}) = \frac{7}{10}$$

$$P\ (\text{Hiking} = \text{Yes}) = \frac{3}{10}$$

The probability of each category in every feature was calculated regardless of the target label:

Following the rule: $\frac{\textbf{\textit{Number of rows in this category of the feature}}}{\textbf{\textit{Number of rows in the dataset}}}$

Starting with Temperature:

$P(\text{Temperature} = \text{Hot}) = \frac{3}{10}$

$P(\text{Temperature} = \text{Mild}) = \frac{4}{10}$

$P(\text{Temperature} = \text{Cool}) = \frac{3}{10}$

Then Humidity:

$P(\text{Humidity} = \text{High}) = \frac{6}{10}$

$P(\text{Humidity} = \text{Normal}) = \frac{4}{10}$

Then Wind:

$P(\text{Wind} = \text{Strong}) = \frac{6}{10}$

$P(\text{Wind} = \text{Weak}) = \frac{4}{10}$

And Lastly Weather:

$P(\text{Weather} = \text{Sunny}) = \frac{4}{10}$

$P(\text{Weather} = \text{Cloudy}) = \frac{3}{10}$

$P(\text{Weather} = \text{Rainy}) = \frac{3}{10}$

The conditional probability of every category of each feature as well as the gini index for it, the weighted gini index for each feature, and for the whole dataset using the following rules:

$$\textbf{GINI (Dataset)} = \mathbf{1} - \left(\frac{\textit{Number of rows where label=Yes}}{\textit{Number of rows of the dataset}}\right)^{\mathbf{2}} - \left(\frac{\textit{Number of rows where label=No}}{\textit{Number of rows of the dataset}}\right)^{\mathbf{2}}$$

$$= \mathbf{1} - \frac{\mathbf{3}}{\mathbf{10}}^{\mathbf{2}} - \frac{\mathbf{7}}{\mathbf{10}}^{\mathbf{2}} = \mathbf{0.42}$$

Conditional Probability =

$$\frac{\textit{Joint probability where a certain label and a certain category of a feature } (\boldsymbol{P}(\boldsymbol{A} \textbf{ and } \boldsymbol{B}))}{\textit{Probability of certain category of a feature } (\boldsymbol{P}(\boldsymbol{B}))}$$

$$\textbf{Gini (Node)} = \mathbf{1} - \sum_j [\boldsymbol{P}(\boldsymbol{j}|\boldsymbol{Node})]^{\mathbf{2}}$$

	Temperature =Hot	Temperature =Mild	Temperature =Cool	Humidity =High	Humidity =Normal
Yes	$\frac{1}{3}$	$\frac{2}{4}$	$\frac{0}{3}$	$\frac{2}{6}$	$\frac{1}{4}$
No	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{3}{3}$	$\frac{4}{6}$	$\frac{3}{4}$
Gini (category)	$1 - \frac{1^2}{3} - \frac{2^2}{3} = 0.444$	$1 - \frac{2^2}{4} - \frac{2^2}{4} = 0.5$	$1 - \frac{0^2}{3} - \frac{3^2}{3} = 0$	$1 - \frac{2^2}{6} - \frac{4^2}{6} = 0.444$	$1 - \frac{1^2}{4} - \frac{3^2}{4} = 0.375$
Weighted Gini	$0.444 * \frac{3}{10} + 0.5 * \frac{4}{10} + 0 * \frac{3}{10} = 0.3332$			$0.444 * \frac{6}{10} + 0.375 * \frac{4}{10} = 0.4164$	

	Wind=Strong	Wind=Weak	Weather =Sunny	Weather =Cloudy	Weather =Rainy
Yes	$\frac{1}{6}$	$\frac{2}{4}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{3}$
No	$\frac{5}{6}$	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{2}{3}$	$\frac{2}{3}$
Gini (category)	$1-\frac{1}{6}^2-\frac{5}{6}^2 = 0.277$	$1-\frac{2}{4}^2-\frac{2}{4}^2 = 0.5$	$1-\frac{1}{4}^2-\frac{3}{4}^2 = 0.375$	$1-\frac{2}{3}^2-\frac{1}{3}^2 = 0.444$	$1-\frac{2}{3}^2-\frac{1}{3}^2 = 0.444$
Weighted Gini	$0.277*\frac{6}{10}+0.5*\frac{4}{10}=0.3662$		$0.375*\frac{4}{10}+0.444*\frac{3}{10}+ 0.444*\frac{3}{10}=0.4164$		

Temperature provides the best split as it is the lowest Gini index.

rows where temperature is cool are pure with 0 Gini index, so they lead to a leaf node (decision) and their lines are excluded from the dataset.

rows where temperature is hot have lower Gini than those where temperature is Mild, so they are explored next.

Weather (F1)	Temperature (F2)	Humidty (F3)	Wind(F4)	Hiking (Labels)
Sunny	Hot	High	Weak	Yes
Sunny	Hot	High	Strong	No
Sunny	Hot	High	Strong	No

The probability of each category in every feature was calculated regardless of the target label:

Following the rule: $\frac{\boldsymbol{Number\ of\ rows\ in\ this\ category\ of\ the\ feature}}{\boldsymbol{Number\ of\ rows\ in\ the\ \ subset}}$

Starting at Humidity:

$\mathrm{P(Humidity = High)} = \frac{3}{3}$

Then Wind:

$\mathrm{P(Wind = Strong)} = \frac{2}{3}$

$\mathrm{P(Wind = Weak)} = \frac{1}{3}$

And Lastly Weather:

$\mathrm{P(Weather = Sunny)} = \frac{3}{3}$

The conditional probability of every category of each feature as well as the gini index for it, the weighted gini index for each feature, and for the whole dataset using the following rules:

GINI (Dataset) =

$$\mathbf{1} - \left(\frac{\boldsymbol{Number\ of\ rows\ where\ label = Yes}}{\boldsymbol{Number\ of\ rows\ of\ the\ subset}}\right)^{\mathbf{2}} - \left(\frac{\boldsymbol{Number\ of\ rows\ where\ label = No}}{\boldsymbol{Number\ of\ rows\ of\ the\ subset}}\right)^{\mathbf{2}}$$

$$= \mathbf{1} - \frac{\mathbf{1}^{\mathbf{2}}}{\mathbf{3}} - \frac{\mathbf{2}^{\mathbf{2}}}{\mathbf{3}} = \mathbf{0.44}$$

Conditional Probability =

$$\frac{\boldsymbol{Joint\ probability\ where\ a\ certain\ label\ and\ a\ certain\ category\ of\ a\ feature}\ (\boldsymbol{P}(\boldsymbol{A\ and\ B}))}{\boldsymbol{Probability\ of\ certain\ category\ of\ a\ feature}\ (\boldsymbol{P}(\boldsymbol{B}))}$$

Gini (Node) = $\mathbf{1} - \sum_{j}[\boldsymbol{P}(\boldsymbol{j}|\boldsymbol{Node})]^{\mathbf{2}}$

	Weather =Sunny	Wind =Strong	Wind =Weak	Humidity =High
Yes	$\frac{1}{3}$	$\frac{0}{2}$	$\frac{1}{1}$	$\frac{1}{3}$
0No	$\frac{2}{3}$	$\frac{2}{2}$	$\frac{0}{1}$	$\frac{2}{3}$
Gini (category)	$1 - \frac{1^2}{3} - \frac{2^2}{3} = 0.444$	$1 - \frac{0^2}{2} - \frac{2^2}{2} = 0$	$1 - \frac{0^2}{1} - \frac{1^2}{1} = 0$	$1 - \frac{1^2}{3} - \frac{2^2}{3} = 0.444$

Weighted Gini	$0.444 * \frac{3}{3} = 0.444$	$0 * \frac{2}{3} + 0 * \frac{1}{3} = 0$	$0.444 * \frac{3}{3} = 0.444$

Wind provides the best split as it is the lowest Gini index.

rows where Wind is Strong or Weak are pure with 0 Gini index, so they lead to a leaf node (decision) and their lines are excluded from the dataset.

rows where temperature is Mild have the next lower Gini, so they are explored next.

Weather (F1)	Temperature (F2)	Humidty (F3)	Wind(F4)	Hiking (Labels)
Rainy	Mild	Normal	Strong	Yes
Cloudy	Mild	High	Strong	No
Sunny	Mild	High	Strong	No
Cloudy	Mild	High	Weak	Yes

The probability of each category in every feature was calculated regardless of the target label:

Following the rule: $\frac{\boldsymbol{Number\ of\ rows\ in\ this\ category\ of\ the\ feature}}{\boldsymbol{Number\ of\ rows\ in\ the\ subset}}$

Starting at Humidity:

$$\mathrm{P(Humidity = High)} = \frac{3}{4}$$

$$\mathrm{P(Humidity = Normal)} = \frac{1}{4}$$

Then Wind:

$$\mathrm{P(Wind = Strong)} = \frac{3}{4}$$

$$\mathrm{P(Wind = Weak)} = \frac{1}{4}$$

And Lastly Weather:

$$P(\text{Weather} = \text{Sunny}) = \frac{1}{4}$$

$$P(\text{Weather} = \text{Cloudy}) = \frac{2}{4}$$

$$P(\text{Weather} = \text{Rainy}) = \frac{1}{4}$$

The conditional probability of every category of each feature as well as the gini index for it, the weighted gini index for each feature, and for the whole dataset using the following rules:

GINI (Dataset) =

$$\mathbf{1} - \left(\frac{\textit{Number of rows where label} = \textit{Yes}}{\textit{Number of rows of the subset}}\right)^{2} - \left(\frac{\textit{Number of rows where label} = \textit{No}}{\textit{Number of rows of the subset}}\right)^{2}$$

$$= \mathbf{1} - \frac{2^2}{4} - \frac{2^2}{4} = \mathbf{0.5}$$

Conditional Probability =

$$\frac{\textit{Joint probability where a certain label and a certain category of a feature } (P(A \text{ and } B))}{\textit{Probability of certain category of a feature } (P(B))}$$

Gini (Node) = $\mathbf{1} - \sum_j [P(j|Node)]^2$

	Wind=Strong	Wind=Weak	Weather =Sunny	Weather =Cloudy	Weather =Rainy
Yes	$\frac{1}{3}$	$\frac{1}{1}$	$\frac{0}{1}$	$\frac{1}{2}$	$\frac{1}{1}$
No	$\frac{2}{3}$	$\frac{0}{1}$	$\frac{1}{1}$	$\frac{1}{2}$	$\frac{0}{1}$
Gini (category)	$1 - \frac{1^2}{3} - \frac{2^2}{3} = 0.444$	$1 - \frac{1^2}{1} - \frac{0^2}{1} = 0$	$1 - \frac{1^2}{1} - \frac{0^2}{1} = 0$	$1 - \frac{1^2}{2} - \frac{1^2}{2} = 0.5$	$1 - \frac{1^2}{1} - \frac{0^2}{1} = 0$

Weighted Gini	$0.444 * \frac{3}{4} + 0 * \frac{1}{4} = 0.333$	$0 * \frac{1}{4} + 0.5 * \frac{2}{4} + 0 * \frac{1}{4} = 0.25$

	Humidity =High	Humidity =Normal
Yes	$\frac{1}{3}$	$\frac{1}{1}$
No	$\frac{2}{3}$	$\frac{0}{1}$
Gini (category)	$1 - \frac{1}{3}^2 - \frac{2}{3}^2 = 0.444$	$1 - \frac{1}{1}^2 - \frac{0}{1}^2 = 0$
Weighted Gini	$0.444 * \frac{3}{4} + 0 * \frac{1}{4} = 0.333$	

Weather provides the best split as it is the lowest Gini index.

rows where Weather is Rainy or Sunny are pure with 0 Gini index, so they lead to a leaf node (decision) and their lines are excluded from the dataset.

rows where Weather is Cloudy have the next lower Gini, so they are explored next.

Weather (F1)	Temperature (F2)	Humidty (F3)	Wind(F4)	Hiking (Labels)
Cloudy	Mild	High	Strong	No
Cloudy	Mild	High	Weak	Yes

The probability of each category in every feature was calculated regardless of the target label:

Following the rule: $\frac{Number\ of\ rows\ in\ this\ category\ of\ the\ feature}{Number\ of\ rows\ in\ the\ subset}$

Starting with Temperature:

$$P(\text{Temperature} = \text{Mild}) = \frac{2}{2}$$

Then Humidity:

$$P(\text{Humidity} = \text{High}) = \frac{2}{2}$$

Then Wind:

$$P(\text{Wind} = \text{Strong}) = \frac{1}{2}$$

$$P(\text{Wind} = \text{Weak}) = \frac{1}{2}$$

The conditional probability of every category of each feature as well as the gini index for it, the weighted gini index for each feature, and for the whole dataset using the following rules:

GINI (Dataset) =

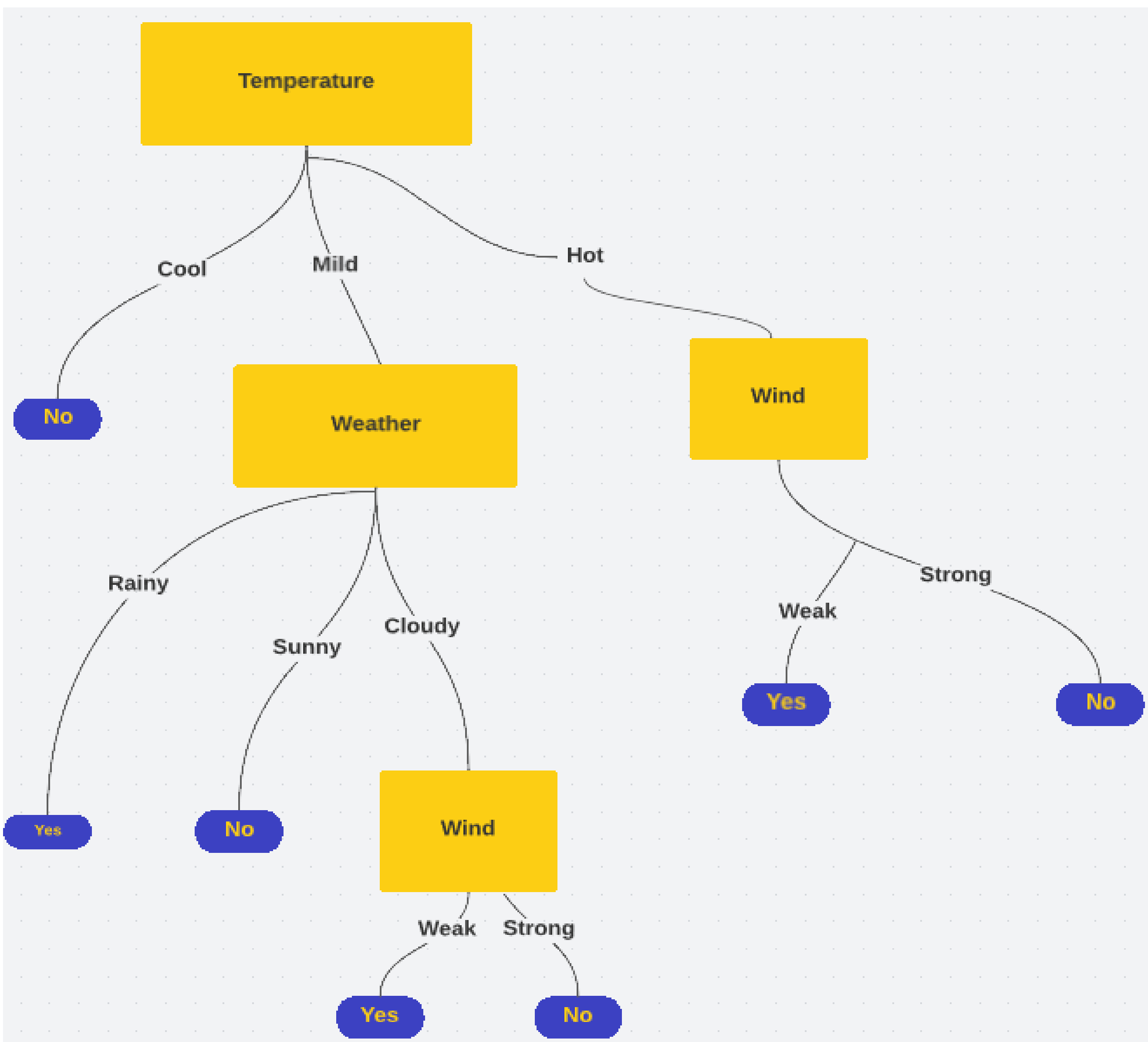
$$\mathbf{1} - \left(\frac{\boldsymbol{Number\ of\ rows\ where\ label = Yes}}{\boldsymbol{Number\ of\ rows\ of\ the\ subset}}\right)^{\mathbf{2}} - \left(\frac{\boldsymbol{Number\ of\ rows\ where\ label = No}}{\boldsymbol{Number\ of\ rows\ of\ the\ subset}}\right)^{\mathbf{2}}$$

$$\mathbf{= 1 - \frac{1}{2}^2 - \frac{1}{2}^2 = 0.5}$$

Conditional Probability =
$\frac{Joint\ probability\ where\ a\ certain\ label\ and\ a\ certain\ category\ of\ a\ feature\ (P(A\ and\ B))}{Probability\ of\ certain\ category\ of\ a\ feature\ (P(B))}$

Gini (Node) = $\mathbf{1} - \sum_j [\boldsymbol{P(j|Node)}]^{\mathbf{2}}$

	Temperature =Mild	Wind =Strong	Wind =Weak	Humidity =High
Yes	$\frac{1}{2}$	$\frac{0}{1}$	$\frac{1}{1}$	$\frac{1}{2}$
No	$\frac{1}{2}$	$\frac{1}{1}$	$\frac{0}{1}$	$\frac{1}{2}$
Gini (category)	$1-\frac{1}{2}^2-\frac{1}{2}^2 = 0.5$	$1-\frac{0}{1}^2-\frac{1}{1}^2=0$	$1-\frac{0}{1}^2-\frac{1}{1}^2=0$	$1-\frac{1}{2}^2-\frac{1}{2}^2=0.5$
Weighted Gini	$0.5*\frac{2}{2}=0.5$	$0*\frac{1}{2}+0*\frac{1}{2}=0$		$0.5*\frac{2}{2}=0.5$



b) **using Information Gain:**

P (Hiking = No) = $\frac{7}{10}$

P (Hiking = Yes) = $\frac{3}{10}$

The probability of each category in every feature was calculated regardless of the target label:

Following the rule: $\frac{\boldsymbol{Number\ of\ rows\ in\ this\ category\ of\ the\ feature}}{\boldsymbol{Number\ of\ rows\ in\ the\ dataset}}$

Starting with Temperature:

P(Temperature = Hot) = $\frac{3}{10}$

P(Temperature = Mild) = $\frac{4}{10}$

P(Temperature = Cool) = $\frac{3}{10}$

Then Humidity:

P(Humidity = High) = $\frac{6}{10}$

P(Humidity = Normal) = $\frac{4}{10}$

Then Wind:

P(Wind = Strong) = $\frac{6}{10}$

P(Wind = Weak) = $\frac{4}{10}$

And Lastly Weather:

P(Weather = Sunny) = $\frac{4}{10}$

P(Weather = Cloudy) = $\frac{3}{10}$

P(Weather = Rainy) = $\frac{3}{10}$

The conditional probability of every category of each feature as well as information gain for each feature, and for the whole dataset using the following rules:

InformationGain (Dataset) =

$$-\frac{\textit{Number of rows where label=Yes}}{\textit{Number of rows of the dataset}} * log_2\left(\frac{\textit{Number of rows where label=Yes}}{\textit{Number of rows of the dataset}}\right) - \frac{\textit{Number of rows where label=No}}{\textit{Number of rows of the dataset}} * log_2\left(\frac{\textit{Number of rows where label=No}}{\textit{Number of rows of the dataset}}\right)$$

$$= -\frac{3}{10}\ log_2\left(\frac{3}{10}\right) - \frac{7}{10}\ log_2\left(\frac{7}{10}\right) = 0.881$$

Conditional Probability =

$$\frac{\textit{Joint probability where a certain label and a certain category of a feature } (P(A \text{ and } B))}{\textit{Probability of certain category of a feature } (P(B))}$$

$$\text{Entropy (Node)} = -\sum_j [P(j|Node)] * log_2(P(j|Node))$$

	Temperature =Hot	Temperature =Mild	Temperature =Cool	Humidity =High	Humidity =Normal
Yes	$\frac{1}{3}$	$\frac{2}{4}$	$\frac{0}{3}$	$\frac{2}{6}$	$\frac{1}{4}$
No	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{3}{3}$	$\frac{4}{6}$	$\frac{3}{4}$
Information Gain (category)	$\frac{3}{10} * (-\frac{1}{3} * \log_2\left(\frac{1}{3}\right) - \frac{2}{3} * \log_2\left(\frac{2}{3}\right)) = 0.275$	$\frac{4}{10} * (-\frac{2}{4} * \log_2\left(\frac{2}{4}\right) - \frac{2}{4} * \log_2\left(\frac{2}{4}\right)) = 0.4$	$\frac{3}{10} * (-0 - \frac{3}{3} * \log_2\left(\frac{3}{3}\right)) = 0$	$\frac{6}{10} * \left(-\frac{2}{6} * \log_2\left(\frac{2}{6}\right) - \frac{4}{6} * \log_2\left(\frac{4}{6}\right)\right) = 0.551$	$\frac{4}{10} * (-\frac{1}{4} * \log_2\left(\frac{1}{4}\right) - \frac{3}{4} * \log_2\left(\frac{3}{4}\right)) = 0.324$
Information Gain	$0.881 - 0.275 - 0 - 0.4 = 0.206$			0.881-0.551-0.324 = 0.006	

	Wind=Strong	Wind=Weak	Weather =Sunny	Weather =Cloudy	Weather =Rainy
Yes	$\frac{1}{6}$	$\frac{2}{4}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{3}$
No	$\frac{5}{6}$	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{2}{3}$	$\frac{2}{3}$
Information Gain (category)	$\frac{6}{10} * (-\frac{1}{6} * \log_2\left(\frac{1}{6}\right) - \frac{5}{6} * \log_2\left(\frac{5}{6}\right)) = 0.39$	$\frac{4}{10} * (-\frac{2}{4} * \log_2\left(\frac{2}{4}\right) - \frac{2}{4} * \log_2\left(\frac{2}{4}\right))$ = 0.4	$\frac{4}{10} * (-\frac{1}{4} * \log_2\left(\frac{1}{4}\right) - \frac{3}{4} * \log_2\left(\frac{3}{4}\right))$ =0.324	$\frac{3}{10} * (-\frac{1}{3} * \log_2\left(\frac{1}{3}\right) - \frac{2}{3} * \log_2\left(\frac{2}{3}\right))$ $=0.275$	$\frac{3}{10} * (-\frac{1}{3} * \log_2\left(\frac{1}{3}\right) - \frac{2}{3} * \log_2\left(\frac{2}{3}\right))$ $=0.275$
Information Gain	$0.881 - 0.39 - 0.4 = 0.091$		$0.881 - 0.324 - 0.275 - 0.275 = 0.097$		

c)

Temperature provides the best split as it is the Highest Information Gain

Weather (F1)	Temperature (F2)	Humidty (F3)	Wind(F4)	Hiking
Sunny	Hot	High	Weak	Yes
Sunny	Hot	High	Strong	No
Sunny	Hot	High	Strong	No

$\text{P (Yes)} = \frac{1}{3}$,

$\text{P (No)} = \frac{2}{3}$

The probability of each category in every feature was calculated regardless of the target label:

Following the rule: $\frac{Number\ of\ rows\ in\ this\ category\ of\ the\ feature}{Number\ of\ rows\ in\ the\ \ subset}$

Starting at Humidity:

P(Humidity = High) = $\frac{3}{3}$

Then Wind:

P(Wind = Strong) =$\frac{2}{3}$

P(Wind = Weak) = $\frac{1}{3}$

And Lastly Weather:

P(Weather = Sunny) = $\frac{3}{3}$

The conditional probability of every category of each feature as well as information gain for each feature, and for the whole dataset using the following rules:

InformationGain (Dataset) =

$$-\frac{Number\ of\ rows\ where\ label{=}Yes}{Number\ of\ rows\ of\ the\ dataset} * log_2\left(\frac{Number\ of\ rows\ where\ label{=}Yes}{Number\ of\ rows\ of\ the\ dataset}\right) - \frac{Number\ of\ rows\ where\ label{=}No}{Number\ of\ rows\ of\ the\ dataset} * log_2\left(\frac{Number\ of\ rows\ where\ label{=}No}{Number\ of\ rows\ of\ the\ dataset}\right)$$

$$= -\frac{1}{3}\ log_2(\frac{1}{3}) - \frac{2}{3}\ log_2(\frac{2}{3}) = \boxed{0.918}$$

Conditional Probability =

$$\frac{Joint\ probability\ where\ a\ certain\ label\ and\ a\ certain\ category\ of\ a\ feature\ (P(A\ and\ B))}{Probability\ of\ certain\ category\ of\ a\ feature\ (P(B))}$$

Entropy (Node) = $-\sum_j[P(j|Node)] * log_2(\ P(j|Node))$

	Weather =Sunny	Wind =Strong	Wind =Weak	Humidity =High
Yes	$\frac{1}{3}$	$\frac{0}{2}$	$\frac{1}{1}$	$\frac{1}{3}$
No	$\frac{2}{3}$	$\frac{2}{2}$	$\frac{0}{1}$	$\frac{2}{3}$
Information Gain (category)	$\frac{3}{3}$(- $\frac{1}{3}$ $log_2(\frac{1}{3})$ - $\frac{2}{3}$ $log_2(\frac{2}{3})$) =0.918	$\frac{2}{3}$(- $\frac{2}{2}$ $log_2(\frac{2}{2})$) =0	$\frac{1}{3}$(- $\frac{1}{1}$ $log_2(\frac{1}{1})$) = 0	$\frac{3}{3}$(- $\frac{1}{3}$ $log_2(\frac{1}{3})$ - $\frac{2}{3}$ $log_2(\frac{2}{3})$) =0.918
Information Gain	$0.918 - 0.918 = 0$	0.918 – 0 -0 = 0.918		$0.918 - 0.918 = 0$

Wind provides the best split as it is the highest information gain.

Weather (F1)	Temperature (F2)	Humidty (F3)	Wind(F4)	Hiking
Rainy	Mild	Normal	Strong	Yes
Cloudy	Mild	High	Strong	No
Sunny	Mild	High	Strong	No
Cloudy	Mild	High	Weak	Yes

P (Yes) = $\frac{2}{4}$,

P (No) = $\frac{2}{4}$

The probability of each category in every feature was calculated regardless of the target label:

Following the rule: $\frac{\boldsymbol{Number\ of\ rows\ in\ this\ category\ of\ the\ feature}}{\boldsymbol{Number\ of\ rows\ in\ the\ subset}}$

Starting at Humidity:

P(Humidity = High) = $\frac{3}{4}$

P(Humidity = Normal) = $\frac{1}{4}$

Then Wind:

P(Wind = Strong) = $\frac{3}{4}$

P(Wind = Weak) = $\frac{1}{4}$

And Lastly Weather:

P(Weather = Sunny) = $\frac{1}{4}$

P(Weather = Cloudy) = $\frac{2}{4}$

P(Weather = Rainy) = $\frac{1}{4}$

The conditional probability of every category of each feature as well as information gain for each feature, and for the whole dataset using the following rules:

InformationGain (Dataset) =

$$-\frac{\textbf{\textit{Number of rows where label=Yes}}}{\textbf{\textit{Number of rows of the dataset}}} * log_2\left(\frac{\textbf{\textit{Number of rows where label=Yes}}}{\textbf{\textit{Number of rows of the dataset}}}\right) - \frac{\textbf{\textit{Number of rows where label=No}}}{\textbf{\textit{Number of rows of the dataset}}} * log_2\left(\frac{\textbf{\textit{Number of rows where label=No}}}{\textbf{\textit{Number of rows of the dataset}}}\right)$$

$$= -\frac{2}{4}\ log_2(\frac{2}{4}) - \frac{2}{4}\ log_2(\frac{2}{4}) = 1$$

Conditional Probability =

$$\frac{\textbf{\textit{Joint probability where a certain label and a certain category of a feature}}\ (\boldsymbol{P}(\boldsymbol{A\ and\ B}))}{\textbf{\textit{Probability of certain category of a feature}}\ (\boldsymbol{P}(\boldsymbol{B}))}$$

Entropy (Node) = $-\sum_j[\boldsymbol{P}(\boldsymbol{j}|\boldsymbol{Node})] * log_2(\ \boldsymbol{P}(\boldsymbol{j}|\boldsymbol{Node}))$

	Wind=Strong	Wind=Weak	Weather =Sunny	Weather =Cloudy	Weather =Rainy
Yes	$\frac{1}{3}$	$\frac{1}{1}$	$\frac{0}{1}$	$\frac{1}{2}$	$\frac{1}{1}$
No	$\frac{2}{3}$	$\frac{0}{1}$	$\frac{1}{1}$	$\frac{1}{2}$	$\frac{0}{1}$
Information Gain (category)	$\frac{3}{4}$ (- $\frac{1}{3}$ $log_2(\frac{1}{3})$ - $\frac{2}{3}$ $log_2(\frac{2}{3})$) = 0.612	$\frac{1}{4}$ (- $\frac{1}{1}$ $log_2(\frac{1}{1})$) = 0	$\frac{1}{4}$ (- $\frac{1}{1}$ $log_2(\frac{1}{1})$) =0	$\frac{2}{4}$ (- $\frac{1}{2}$ $log_2(\frac{1}{2})$ - $\frac{1}{2}$ $log_2(\frac{1}{2})$) =0.5	$\frac{1}{4}$ (- $\frac{1}{1}$ $log_2(\frac{1}{1})$) =0
Information Gain	1 - 0 - 0.612 = 0.388		$1 - 0 - 0.5 - 0 = 0.5$		

	Humidity =High	Humidity =Normal
Yes	$\frac{1}{3}$	$\frac{1}{1}$
No	$\frac{2}{3}$	$\frac{0}{1}$
Information Gain (category)	$\frac{3}{4}$ (- $\frac{1}{3}$ $log_2(\frac{1}{3})$ - $\frac{2}{3}$ $log_2(\frac{2}{3})$) = 0.612	$\frac{1}{4}$ (- $\frac{1}{1}$ $log_2(\frac{1}{1})$) = 0
Information Gain	$1 - 0.612 - 0 = 0.388$	

Weather is the best split for Highest information Gain

Weather (F1)	Temperature (F2)	Humidty (F3)	Wind(F4)	Hiking
Cloudy	Mild	High	Strong	No
Cloudy	Mild	High	Weak	Yes

$\text{P (Yes)} = \frac{1}{2} \quad \text{P (No)} = \frac{1}{2}$

The probability of each category in every feature was calculated regardless of the target label:

Following the rule: $\frac{Number\ of\ rows\ in\ this\ category\ of\ the\ feature}{Number\ of\ rows\ in\ the\ subset}$

Starting with Temperature:

P(Temperature = Mild) = $\frac{2}{2}$

Then Humidity:

P(Humidity = High) = $\frac{2}{2}$

Then Wind:

P(Wind = Strong) = $\frac{1}{2}$

P(Wind = Weak) = $\frac{1}{2}$

The conditional probability of every category of each feature as well as information gain for each feature, and for the whole dataset using the following rules:

InformationGain (Dataset) =

$$-\frac{Number\ of\ rows\ where\ label{=}Yes}{Number\ of\ rows\ of\ the\ dataset} * log_2\left(\frac{Number\ of\ rows\ where\ label{=}Yes}{Number\ of\ rows\ of\ the\ dataset}\right) - \frac{Number\ of\ rows\ where\ label{=}No}{Number\ of\ rows\ of\ the\ dataset} * log_2\left(\frac{Number\ of\ rows\ where\ label{=}No}{Number\ of\ rows\ of\ the\ dataset}\right)$$

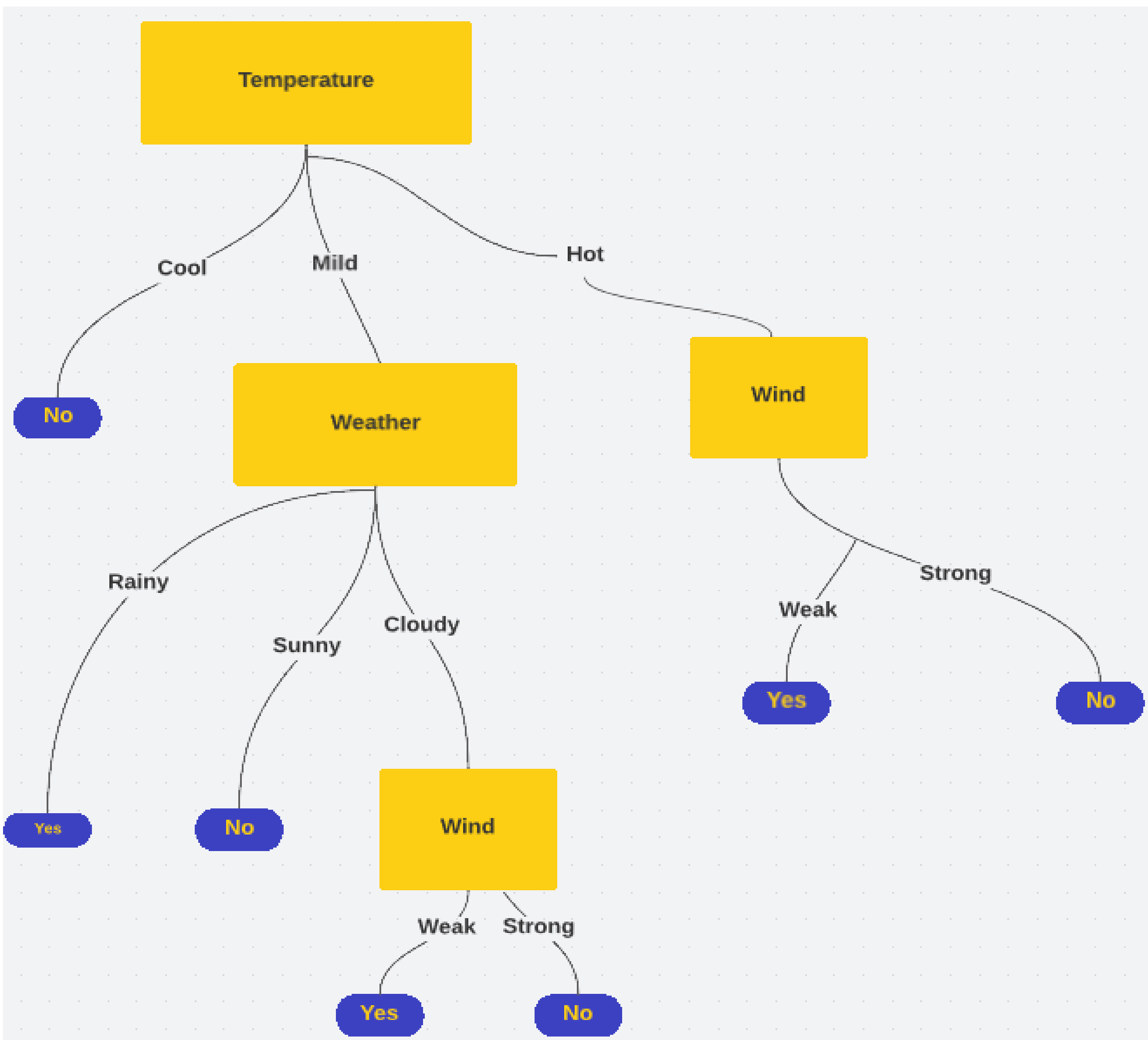
$$= = -\frac{1}{2}\ log_2\left(\frac{1}{2}\right) - \frac{1}{2}\ log_2\left(\frac{1}{2}\right) = 1$$

Conditional Probability =

$$\frac{Joint\ probability\ where\ a\ certain\ label\ and\ a\ certain\ category\ of\ a\ feature\ (P(A\ and\ B))}{Probability\ of\ certain\ category\ of\ a\ feature\ (P(B))}$$

Entropy (Node) = $-\sum_j [P(j|Node)] * log_2(\ P(j|Node))$

	Temperature =Mild	Wind =Strong	Wind =Weak	Humidity =High
Yes	$\frac{1}{2}$	$\frac{0}{1}$	$\frac{1}{1}$	$\frac{1}{2}$
No	$\frac{1}{2}$	$\frac{1}{1}$	$\frac{0}{1}$	$\frac{1}{2}$
Information Gain (category)	$\frac{2}{2}$ (- $\frac{1}{2}$ $log_2\left(\frac{1}{2}\right)$ - $\frac{1}{2}$ $log_2(\frac{1}{2})$) = 1	1 - $\frac{1}{2}$ (- $\frac{1}{1}$ $log_2\left(\frac{1}{1}\right)$) = 0	1 - $\frac{1}{2}$ (- $\frac{1}{1}$ $log_2\left(\frac{1}{1}\right)$) = 0	$\frac{2}{2}$ (- $\frac{1}{2}$ $log_2\left(\frac{1}{2}\right)$ - $\frac{1}{2}$ $log_2(\frac{1}{2})$) = 1
Information Gain	$1-1=0$	$1-0-0=1$		$1-1=0$



2. Coding Part

Functions were made to facilitate such process like function that reads data and splits into train and test, function that gets accuracy, and function that gets prediction:

```
[2]  train_data=pd.read_csv("/content/pendigits-tra.csv",header=None)
     test_data=pd.read_csv("/content/pendigits-tes.csv",header=None)

[3]  X_train=train_data.iloc[:,:-1]
     y_train=train_data.iloc[:,-1]

[4]  X_test=test_data.iloc[:,:-1]
     y_test=test_data.iloc[:,-1]
```

```
def get_accuracies(y_actual, y_predict):
  from sklearn.metrics import classification_report, ConfusionMatrixDisplay, accuracy_score, confusion_matrix
  print('\nClassification Report:\n')
  print(classification_report(y_actual, y_predict))
  cm = confusion_matrix(y_actual, y_predict)
  print('\nAccuracy Score:\n')
  print(accuracy_score(y_actual, y_predict))
  print('\Confusion Matrix Display:\n')
  print(ConfusionMatrixDisplay(cm).plot())
```

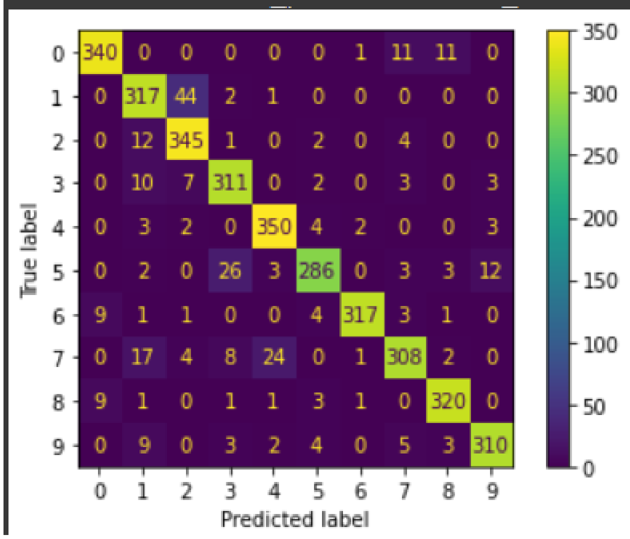
```
[6]  def get_predect(pip,Xtrain,Xtest, ytrain,y):
       pip.fit(Xtrain.values, ytrain.values)

     # Predicting the Test set results
       y_pred = pip.predict(Xtest.values)
       acc=accuracy_score(y, y_pred)*100
       print( acc)
       return y_pred,acc
```

a. Apply Decision Tree:

Decision tree was applied to the dataset and confusion matrix and Classification report were calculated to check the trained model

```
[10] from sklearn.tree import DecisionTreeClassifier
     from sklearn.metrics import accuracy_score
     from sklearn.metrics import classification_report
     from sklearn.metrics import accuracy_score

     estimator = DecisionTreeClassifier(random_state=2022)
     estimator.fit(X_train, y_train)
     DTy_pred = estimator.predict(X_test)
     report = get_accuracies(y_test, DTy_pred)
```

```
Accuracy Score:

0.9159519725557461

Classification Report:

              precision    recall  f1-score   support

           0       0.95      0.94      0.94       363
           1       0.85      0.87      0.86       364
           2       0.86      0.95      0.90       364
           3       0.88      0.93      0.90       336
           4       0.92      0.96      0.94       364
           5       0.94      0.85      0.89       335
           6       0.98      0.94      0.96       336
           7       0.91      0.85      0.88       364
           8       0.94      0.95      0.95       336
           9       0.95      0.92      0.93       336

    accuracy                           0.92      3498
   macro avg       0.92      0.92      0.92      3498
weighted avg       0.92      0.92      0.92      3498
```



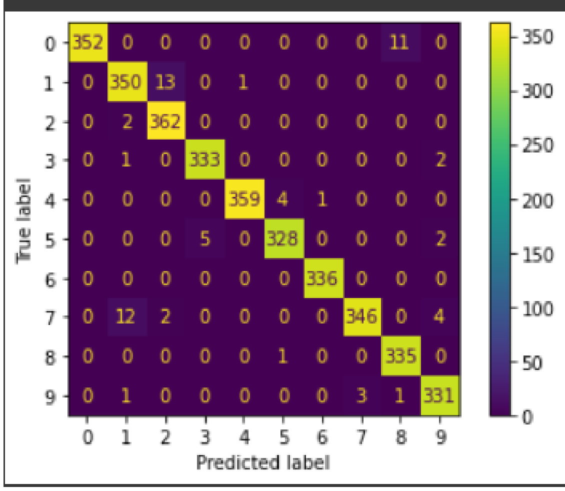
Bagging:

a. Bagging Strategy was applied on both svm and Decision Tree as base estimators and accuracy and confusion matrix were measured for both classifiers

SVM:

```
[11] bag_clf = BaggingClassifier(SVC(), n_estimators=500,bootstrap=True, n_jobs=-1, oob_score=True, random_state=20
     bag_clf.fit(X_train, y_train)
     bag_clf.oob_score_

     0.9958633573525487
```

```
Accuracy Score:

0.9811320754716981

Classification Report:

              precision    recall  f1-score   support

           0       1.00      0.97      0.98       363
           1       0.96      0.96      0.96       364
           2       0.96      0.99      0.98       364
           3       0.99      0.99      0.99       336
           4       1.00      0.99      0.99       364
           5       0.98      0.98      0.98       335
           6       1.00      1.00      1.00       336
           7       0.99      0.95      0.97       364
           8       0.97      1.00      0.98       336
           9       0.98      0.99      0.98       336

    accuracy                           0.98      3498
   macro avg       0.98      0.98      0.98      3498
weighted avg       0.98      0.98      0.98      3498
```

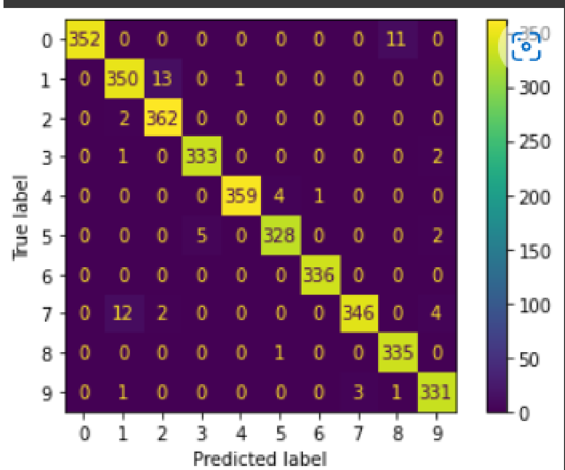


Decision Tree:

```
bag_clf = BaggingClassifier(DecisionTreeClassifier(), n_estimators=500,bootstrap=True, n_jobs=-1, oob_score=True, random_state=2022)
bag_clf.fit(X_train, y_train)
bag_clf.oob_score_
```

```
0.9834534294101949
```

```
Accuracy Score:

0.9811320754716981

Classification Report:

              precision    recall  f1-score   support

           0       1.00      0.97      0.98       363
           1       0.96      0.96      0.96       364
           2       0.96      0.99      0.98       364
           3       0.99      0.99      0.99       336
           4       1.00      0.99      0.99       364
           5       0.98      0.98      0.98       335
           6       1.00      1.00      1.00       336
           7       0.99      0.95      0.97       364
           8       0.97      1.00      0.98       336
           9       0.98      0.99      0.98       336

    accuracy                           0.98      3498
   macro avg       0.98      0.98      0.98      3498
weighted avg       0.98      0.98      0.98      3498
```

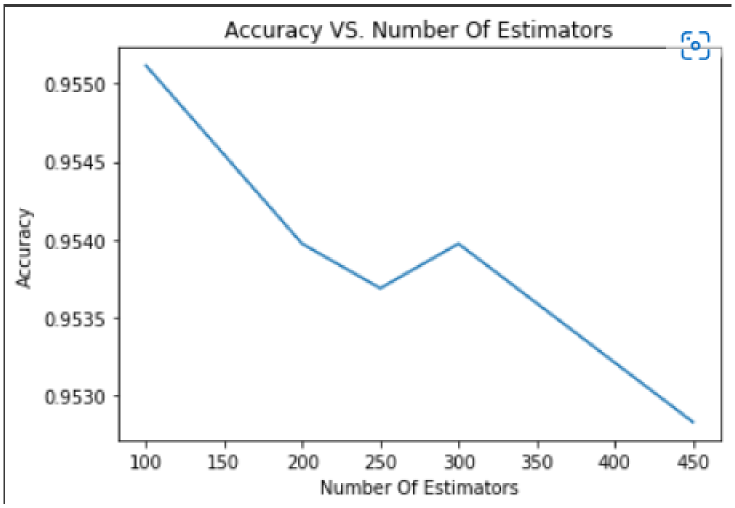


Best Number of Estimators:

Different number of estimators were tried, and they were plotted against the testing accuracy to decide on the best number for decision tree classifier

```
estimators=[100,200,250,300,450]
accuracy=[]
for i in estimators:
  bag_clf = BaggingClassifier(DecisionTreeClassifier(), n_estimators=i,bootstrap=True, n_jobs=-1, oob_score=True, random_state=2
  bag_clf.fit(X_train, y_train)
  y_pred = bag_clf.predict(X_test)
  accuracy.append(accuracy_score(y_test, y_pred))
```

```
plt.plot(estimators,accuracy)
plt.xlabel('Number Of Estimators')
plt.ylabel('Accuracy')
plt.title('Accuracy VS. Number Of Estimators')
plt.show()
```

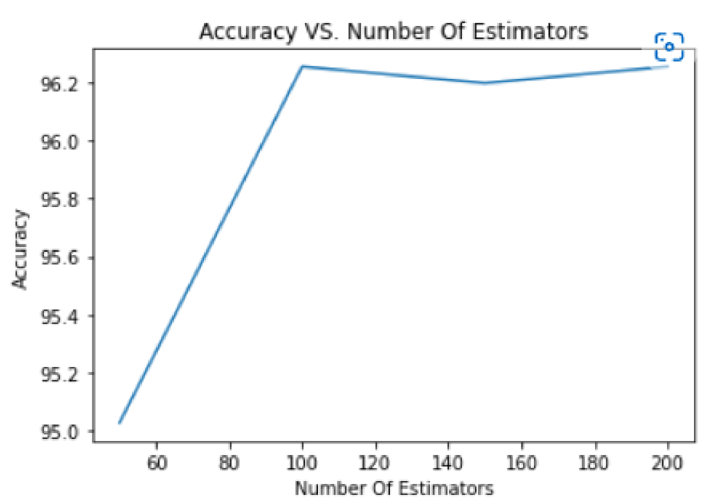


As it is shown from the plot 100 Estimators has managed to get the best testing accuracy over the interval of tried values.

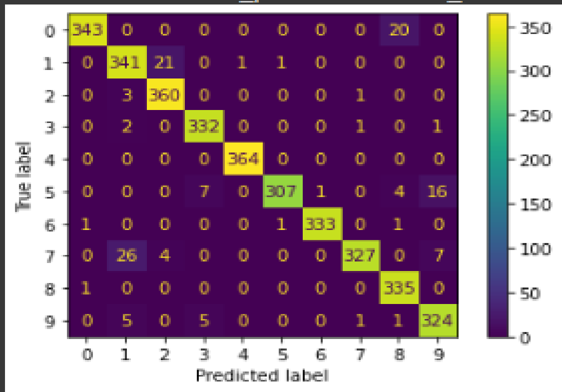
4. Boosting :

A- In the hyper parameter tuning for GradientBoostingClassifier we found that the best value for the number of estimators from the accuracy from these values [50,100,150,200] is both 100 and 200 so we choesd 200

```
[ ] n_estimators=[50,100,150,200]
    accn=[]
    yperdes=[]
    for n in n_estimators:
      model = GradientBoostingClassifier(n_estimators=n)
      y_pred,acc=get_predect(model,X_train, X_test, y_train, y_test)
      accn.append(acc)
      yperdes.append(y_pred)

    95.02572898799315
    96.25500285877644
    96.1978273299028
    96.25500285877644
```



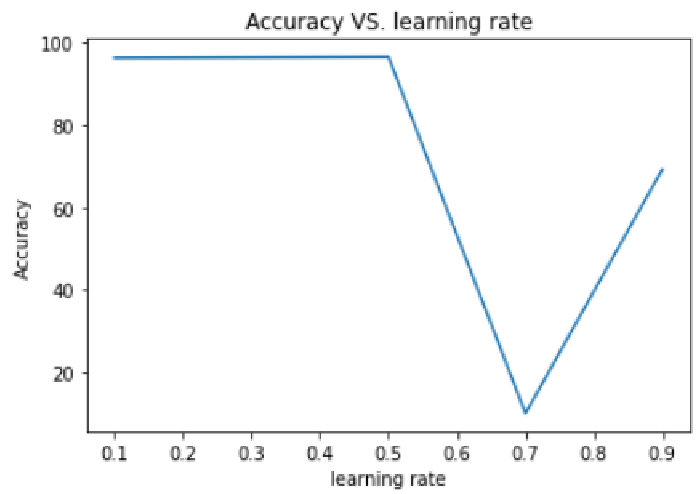
```
0.9622641509433962
\Confusion Matrix Display:

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7f63132b8310>
```



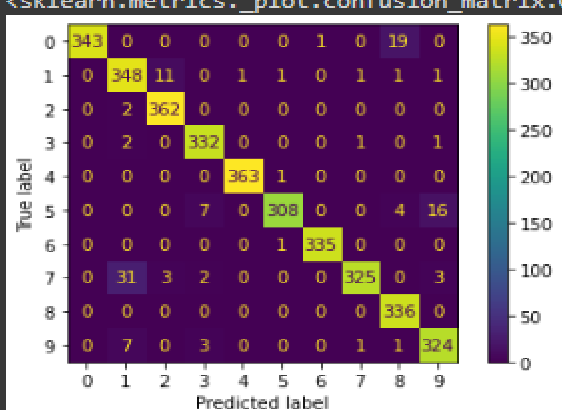
And using the number of estimator as 200 we found that the best value for the learning rat is .5 from these values [.1,.5,.7,.9]

```
LR=[.1,.5,.7,.9]
acclr=[]
yperdeslr=[]

for n in LR:
  model = GradientBoostingClassifier(n_estimators=200,learning_rate=n)
  y_pred,acc=get_predect(model,X_train, X_test, y_train, y_test)
  acclr.append(acc)
  yperdeslr.append(y_pred)
```

```
96.25500285877644
96.51229273870783
10.062893081761008
69.23956546598056
```




```
0.9651229273870783
\Confusion Matrix Display:

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7f63120d4e90>
```

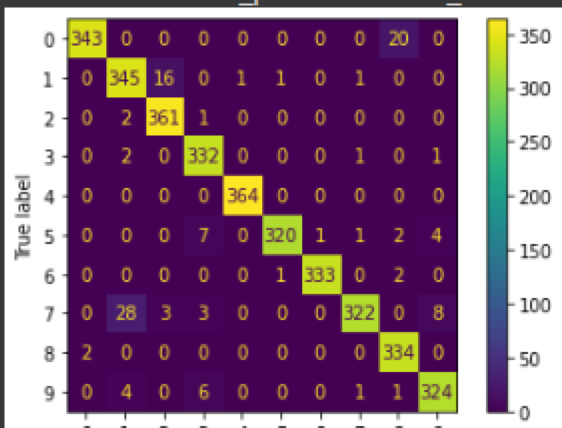


B- Using the best parameter from the 4-A part we build an Xgboost model :

```python
from xgboost import XGBClassifier
xgboost = XGBClassifier(n_estimators = 200, learning_rate = 0.5)
y_pred,acc=get_predect(xgboost,X_train, X_test, y_train, y_test)
get_accuracies(y_test, y_pred)
```

And got the following accuracy and confusion matrix:

```
0.9656946826758147
\Confusion Matrix Display:

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7f6312368e90>
```



C- between the Xgboost and GradientBoostingClassifier the difference is not that big even though the Xgboost is better in term of accuracy ,and between the acuracy and confusion matrix the confusion matrix is better because it lets us know better which class is the model confused about ,the bagging was better than the boosting in this problem in term of accuracy and confusion matrix result