2143488 BIG DATA AND ARTIFICIAL INTELLIGENCE

DECISION TREE

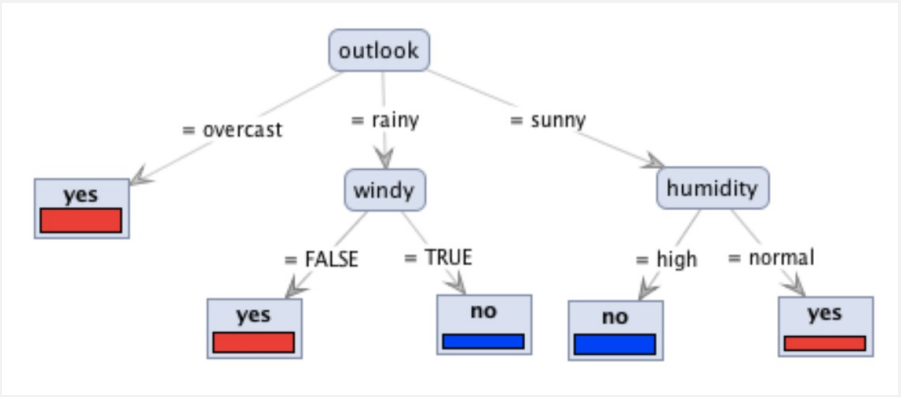
DR. JING TANG

DECISION TREE

- Decision tree creates decision rules from the data.
- It is usually applied for **classification (categorical targets)**, but it can be used for **regression(real number targets** as well. (CART)

Logistic Regression : use linear line to fit
↳ But if you have multiple parameters, it's likely to not be linear.

ID	Outlook	Temperature	Humidity	Windy	Play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	mild	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	mild	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

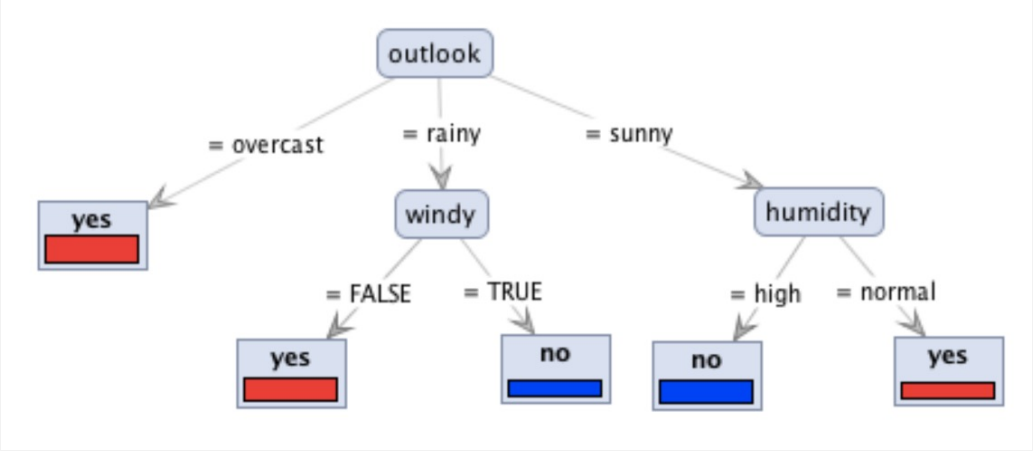


Classification and Regression Trees Leo Breiman, Jerome Friedman, Charles J. Stone, R.A. Olshen (1984)

Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106

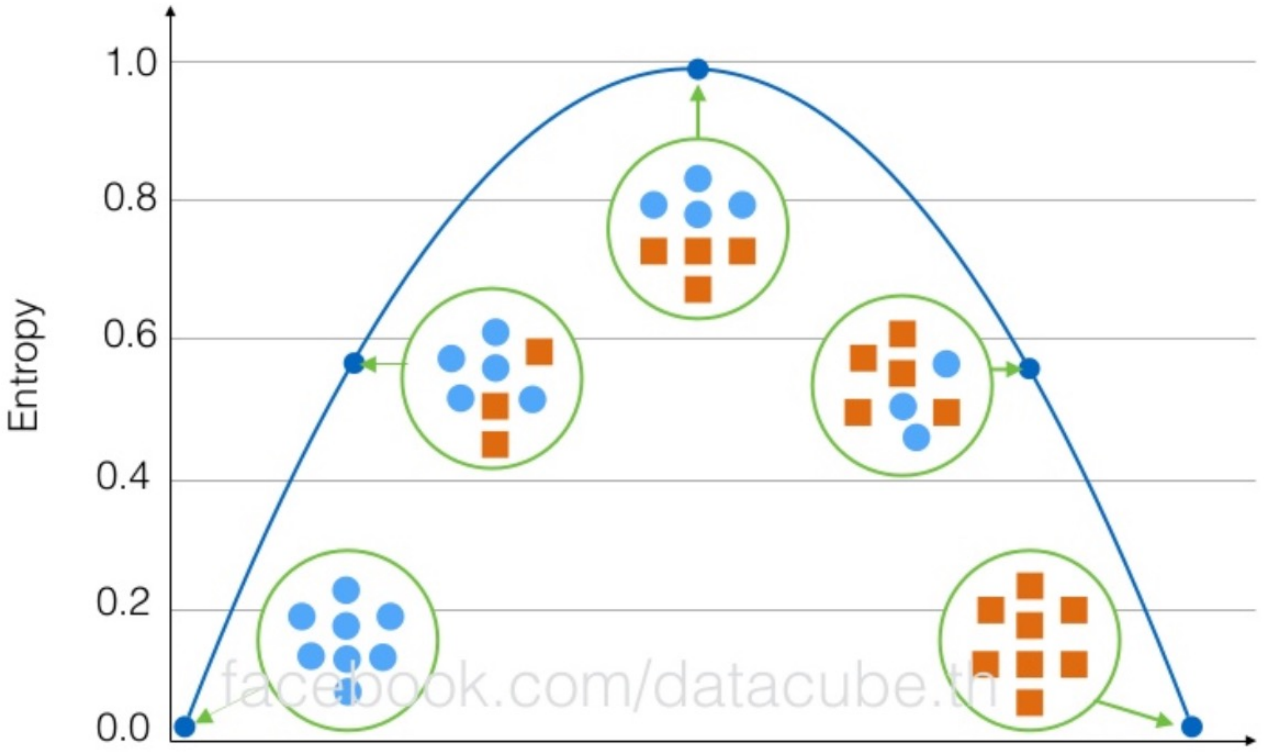
HOW TO BUILD THE DECISION TREE

- How does the model know at what value to **split** the top node?
- How does the model determine the **order of the nodes** from top to bottom?



ENTROPY

- If the entropy equals to **zero**, it means that the data is **homogeneous**.



ENTROPY AND IG 1

- Compute **Entropy** and **Information Gain (IG)**

$$Entropy(c_i) = \sum_{i=1}^{C} -p(c_i)\, log_2\, p(c_i)$$

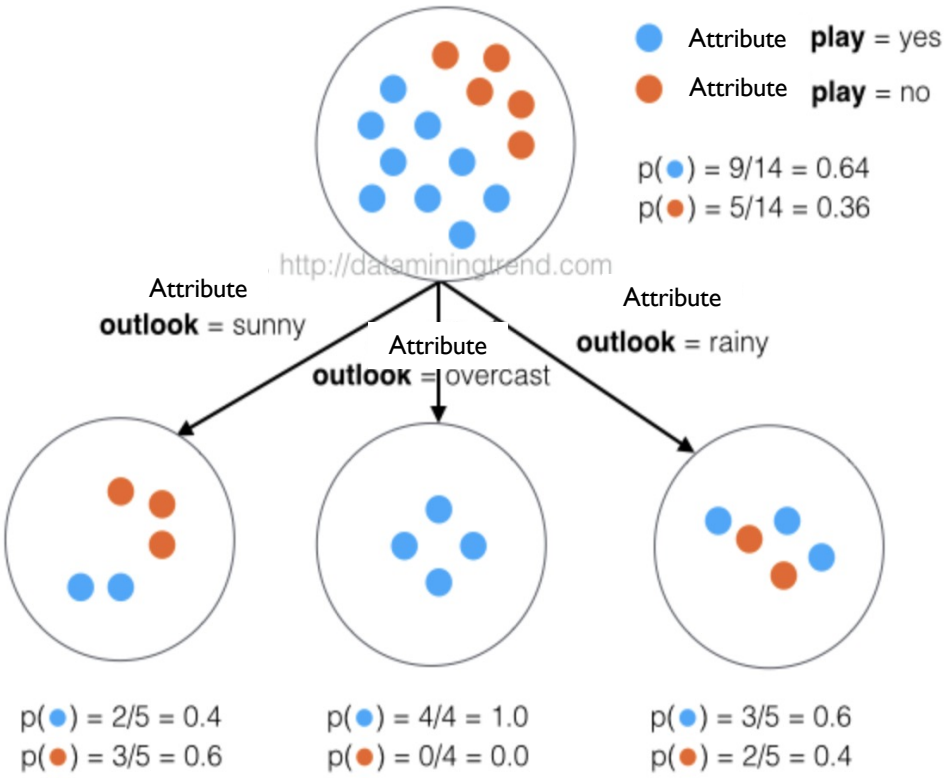
$$IG(parent, child) = Entropy(parent) - \sum_{j=1}^{N} p(c_j) \times Entropy(c_j)$$

Sum of $p(c_j)$ entropy of each children

↑ prob to go to each node

- The attribute which **maximizes the information gain (IG)** the most is selected as the root node or top node.

ENTROPY AND IG 2



entropy (parent) = $-p(\bullet) \times \log_2 p(\bullet) - p(\bullet) \times \log_2 p(\bullet)$
= $-[0.64 \times \log_2(0.64) + 0.36 \times \log_2(0.36)]$
= $-[0.64 \times -0.64 + 0.36 \times -1.47]$
= 0.94

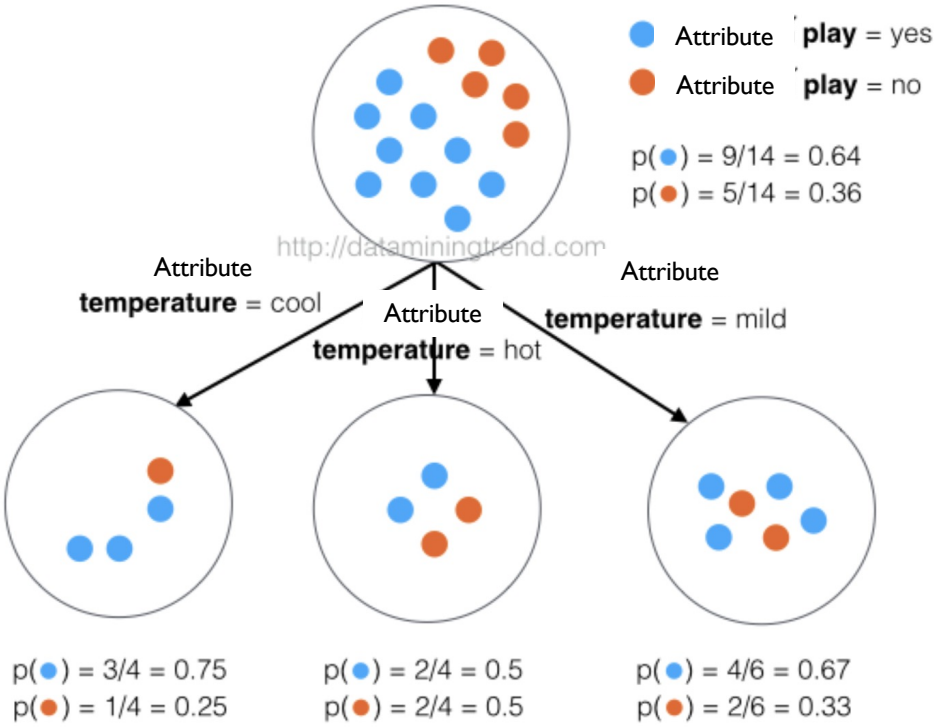
entropy(**outlook** = sunny) = $-p(\bullet) \times \log_2 p(\bullet) - p(\bullet) \times \log_2 p(\bullet)$
= $-[0.4 \times \log_2(0.4) + 0.6 \times \log_2(0.6)]$
= $-[0.4 \times -1.32 + 0.6 \times -0.74]$
= 0.97

entropy(**outlook** = overcast) = $-p(\bullet) \times \log_2 p(\bullet) - p(\bullet) \times \log_2 p(\bullet)$
= $-[1.0 \times \log_2(1.0) + 0 \times \log_2(0)]$
= $-[1.0 \times 0 + 0 \times 1]$
= 0

entropy(**outlook** = rainy) = $-p(\bullet) \times \log_2 p(\bullet) - p(\bullet) \times \log_2 p(\bullet)$
= $-[0.6 \times \log_2(0.6) + 0.4 \times \log_2(0.4)]$
= $-[0.6 \times -0.74 + 0.4 \times -1.32]$
= 0.97

IG(parent, child) = entropy (parent) – [p(**outlook** = sunny) × entropy(**outlook** = sunny)+p(**outlook** = overcast) × entropy(**outlook** = overcast)+p(**outlook** = rainy) × entropy(**outlook** = rainy)]
= 0.94 - [0.36 x 0.97 + 0.29 x 0 + 0.36 x 0.97]
= 0.94 – 0.69
= 0.25

ENTROPY AND IG 3



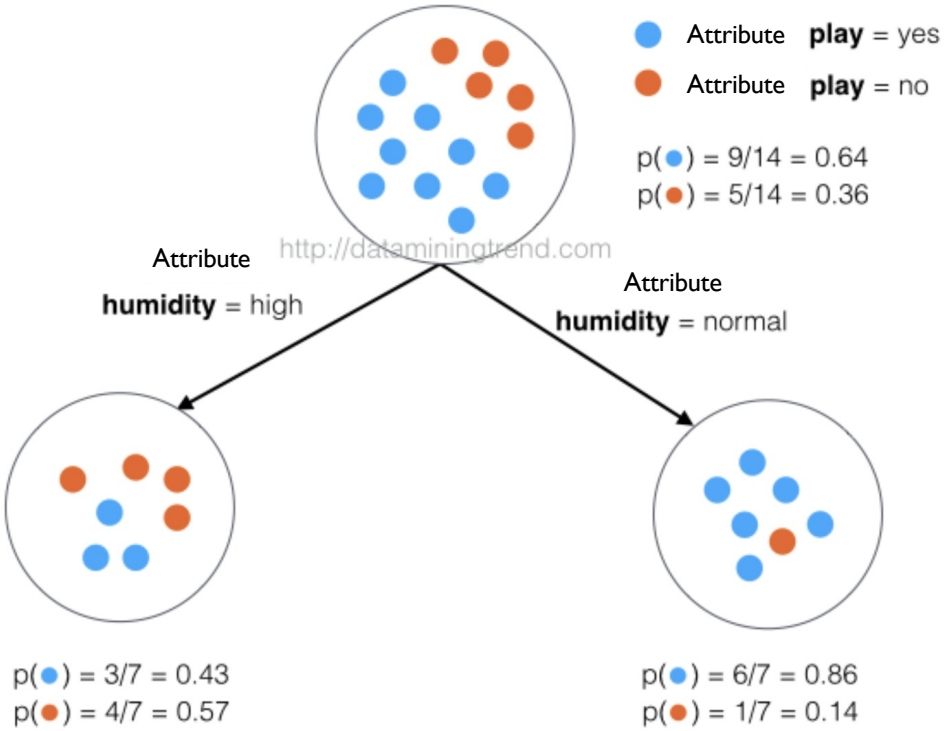
entropy(**temperature** = cool) = -p(•) × $\log_2$ p(•) – p(•) × $\log_2$ p(•)
= -[0.75 × $\log_2$(0.75) + 0.25 × $\log_2$(0.25)]
= 0.81

entropy(**temperature** = hot) = -p(•) × $\log_2$ p(•) – p(•) × $\log_2$ p(•)
= -[0.5 × $\log_2$(0.5) + 0.5× $\log_2$(0.5)]
= 1

entropy(**temperature** = mild) = -p(•) × $\log_2$ p(•) – p(•) × $\log_2$ p(•)
= -[0.67 × $\log_2$(0.67) + 0.33 × $\log_2$(0.33)]
= 0.91

IG(parent, child) = entropy (parent) – [p(**temperature** = cool) × entropy(**temperature** = cool)+p(**temperature** = hot) × entropy(**temperature** = hot)+p(**temperature** = mild) × entropy(**temperature** = mild)]
= 0.94 – [0.29 × 0.81 + 0.29 × 1 + 0.42 × 0.91]
= 0.94 - 0.91
= 0.03

ENTROPY AND IG 4



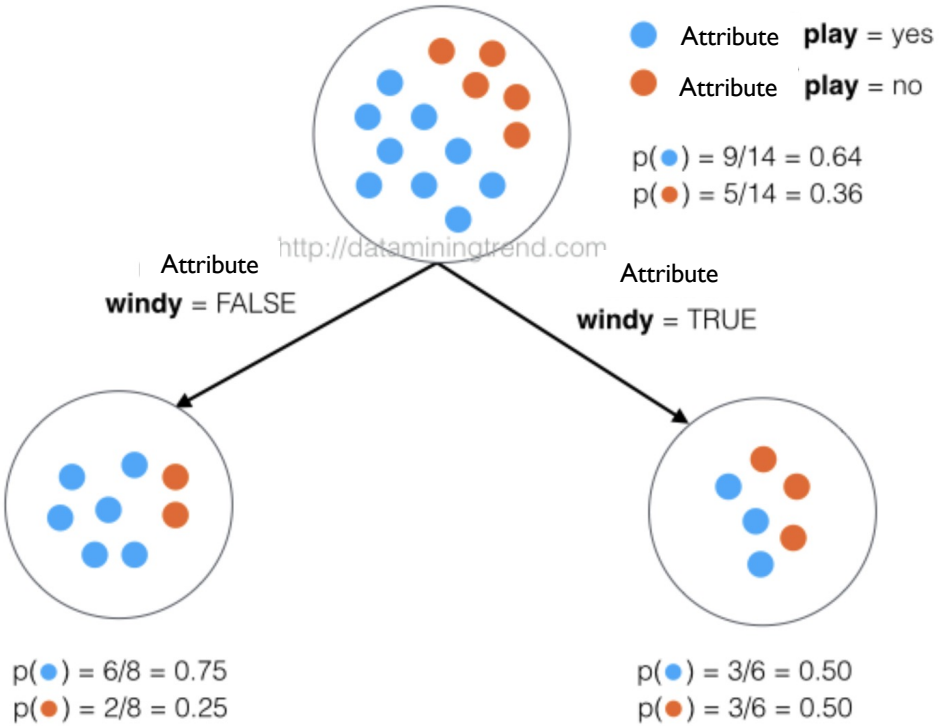
entropy(**humidity** = high) = $-p(\bullet) \times \log_2 p(\bullet) - p(\bullet) \times \log_2 p(\bullet)$
$= -[0.43 \times \log_2(0.43) + 0.57 \times \log_2(0.57)]$
$= 0.99$

entropy(**humidity** = normal) = $-p(\bullet) \times \log_2 p(\bullet) - p(\bullet) \times \log_2 p(\bullet)$
$= -[0.86 \times \log_2(0.86) + 0.14 \times \log_2(0.14)]$
$= 0.58$

IG(parent, child) = entropy (parent) – [p(**humidity** = cool) × entropy(**humidity** = cool)+p(**humidity** = hot) × entropy(**humidity** = hot)]

$= 0.94-[0.5 \times 0.99 + 0.5 \times 0.58]$
$= 0.94 - 0.79$
$= 0.15$

ENTROPY AND IG 5



entropy(**windy** = FALSE) = $-p(\bullet) \times \log_2 p(\bullet) - p(\bullet) \times \log_2 p(\bullet)$

$= -[0.75 \times \log_2(0.75) + 0.25 \times \log_2(0.25)]$

$= 0.81$

entropy(**windy** = TRUE) = $-p(\bullet) \times \log_2 p(\bullet) - p(\bullet) \times \log_2 p(\bullet)$

$= -[0.5 \times \log_2(0.5) + 0.5 \times \log_2(0.5)]$

$= 1$

IG(parent, child) = entropy (parent) – [p(**windy** = FALSE) × entropy(**windy** = FALSE)+p(**windy** = TRUE) × entropy(**windy** = TRUE)]

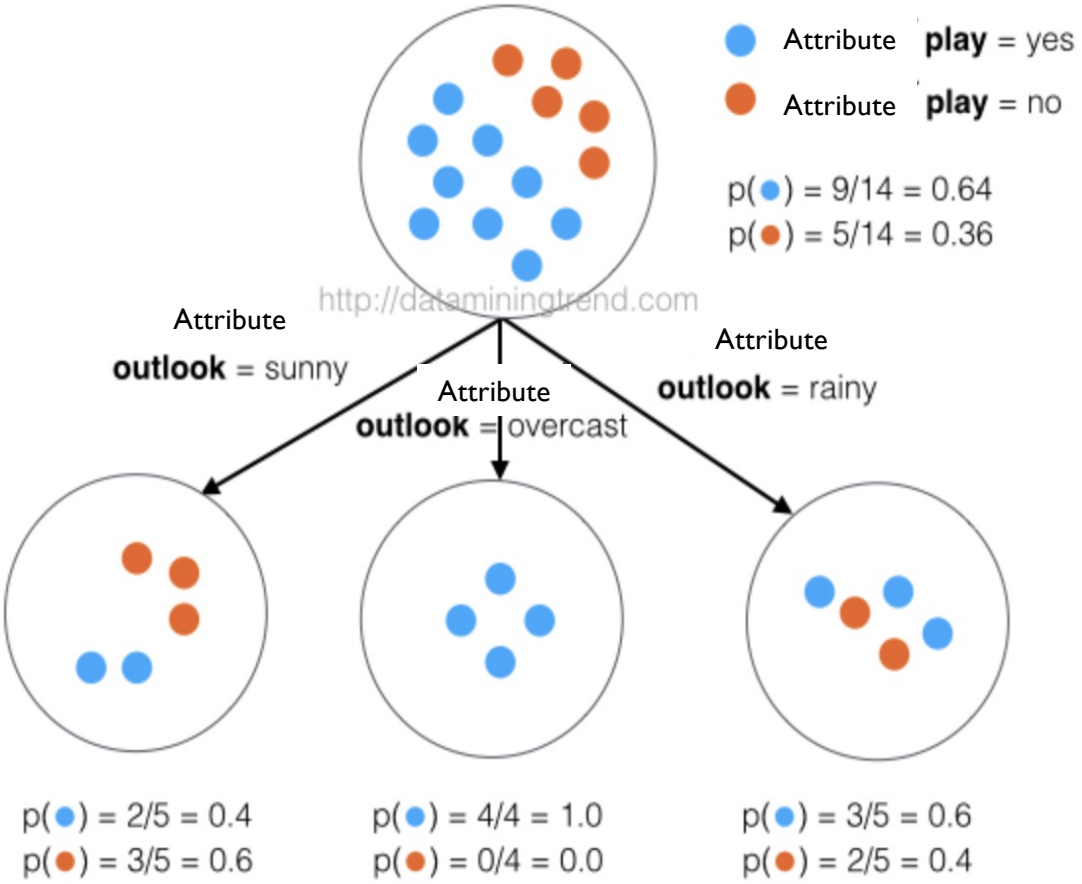
$= 0.94-[0.57 \times 0.81 + 0.43 \times 1]$

$= 0.94 - 0.89$

$= 0.05$

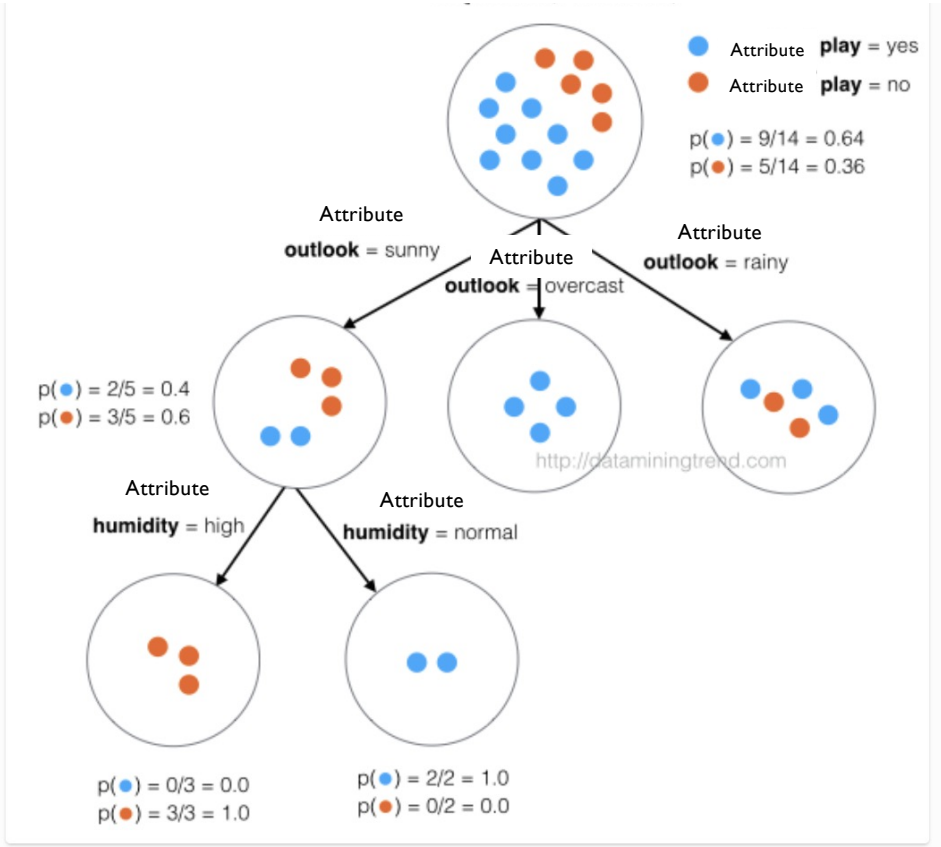
ENTROPY AND IG 6

Attribute	IG
Outlook	**0.25**
Temperature	0.03
Humidity	0.15
Windy	0.05



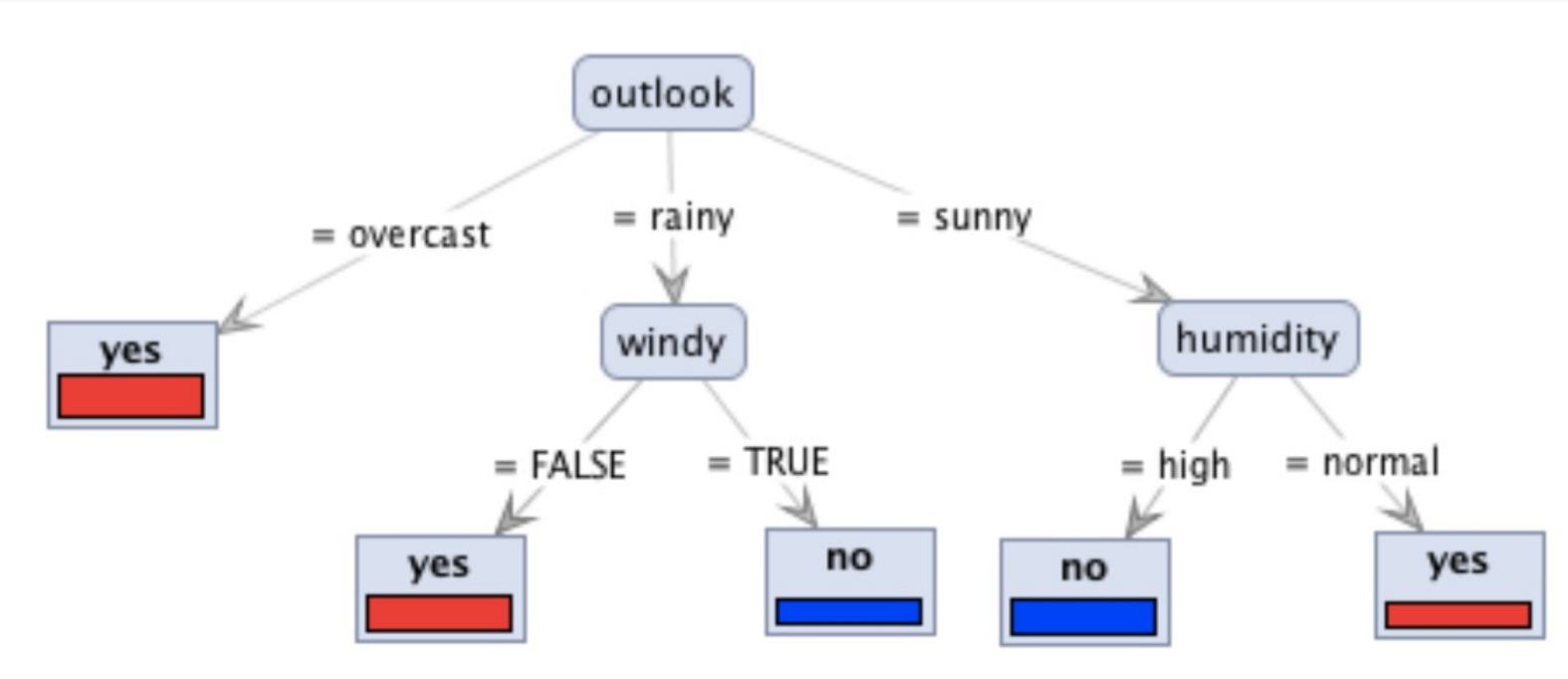
ENTROPY AND IG 7

ID	Outlook	Temperature	Humidity	Windy	Play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	mild	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	mild	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no



SELECT ATTRIBUTE

- Decision tree model after training



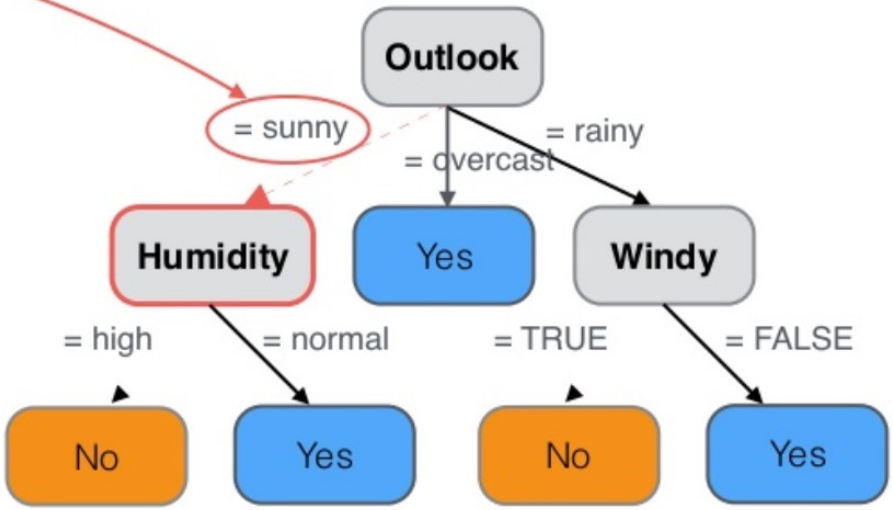
PREDICTION 1

- Use the decision tree model to make a prediction for testing data

ID	Outlook	Temperature	Humidity	Windy
1	sunny	hot	high	FALSE

Testing data

Problem: could overfit



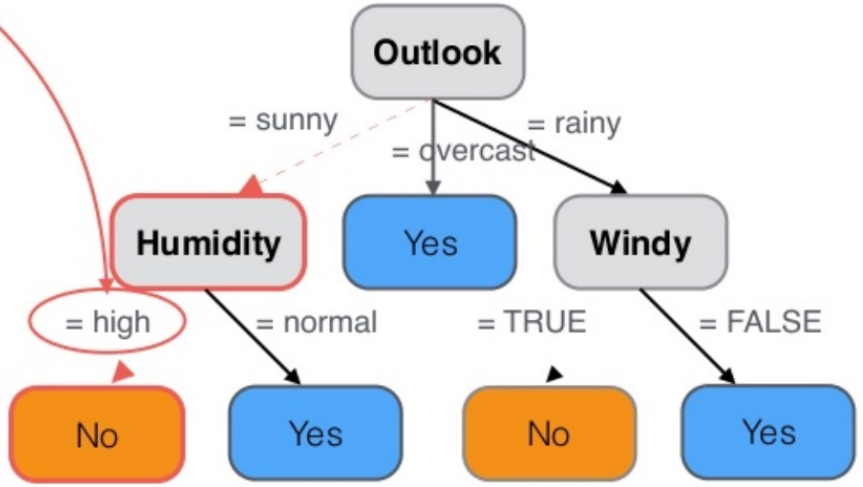
Decision tree model

PREDICTION 2

- Use the decision tree model to make a prediction for testing data

ID	Outlook	Temperature	Humidity	Windy
1	sunny	hot	high	FALSE

Testing data



Decision tree model

NUMERIC ATTRIBUTE 1

- In case that the input is numerical data, we have to transform it into categorical data before constructing decision tree. Logistic Regression: Transfer categorical values to 0 & 1
- Sort the data from **smallest to largest value**
- Separate the data into two parts by a boundary that is an **average** between values of **two adjacent samples**
- Compute **information gain (IG)** of each part of the data
- Choose the boundary giving the **highest IG** for further analysis

↳ cut 1/2 3 4 5 , 12/3 4 5 , 12 3/4 5 ... check cutting where results in the highest information gain

NUMERIC ATTRIBUTE 2

- If humidity = 67.5 is used as the boundary, IG = 0.11.

ID	Humidity	Play
7	65.0	no
6	70.0	no
9	70.0	yes
11	70.0	yes
13	75.0	yes
3	78.0	no
5	80.0	yes
10	80.0	no
14	80.0	yes
1	85.0	yes
2	90.0	yes
12	90.0	yes
8	95.0	yes
4	96.0	no

AVG = 67.5

ID	Humidity	Play
7	≤ 67.5	no
6	> 67.5	no
9	> 67.5	yes
11	> 67.5	yes
13	> 67.5	yes
3	> 67.5	no
5	> 67.5	yes
10	> 67.5	no
14	> 67.5	yes
1	> 67.5	yes
2	> 67.5	yes
12	> 67.5	yes
8	> 67.5	yes
4	> 67.5	no

NUMERIC ATTRIBUTE 3

- If humidity = 72.5 is used as the boundary, IG = 0.25.

ID	Humidity	Play
7	65.0	no
6	70.0	no
9	70.0	yes
11	70.0	yes
13	75.0	yes
3	78.0	no
5	80.0	yes
10	80.0	no
14	80.0	yes
1	85.0	yes
2	90.0	yes
12	90.0	yes
8	95.0	yes
4	96.0	no

Cut here

AVG = 72.5

ID	Humidity	Play
7	≤ 72.5	no
6	≤ 72.5	no
9	≤ 72.5	yes
11	≤ 72.5	yes
13	> 72.5	yes
3	> 72.5	no
5	> 72.5	yes
10	> 72.5	no
14	> 72.5	yes
1	> 72.5	yes
2	> 72.5	yes
12	> 72.5	yes
8	> 72.5	yes
4	> 72.5	no

NUMERIC ATTRIBUTE 4

- If humidity = 76.5 is used as the boundary, IG = 0.03.

ID	Humidity	Play
7	65.0	no
6	70.0	no
9	70.0	yes
11	70.0	yes
13	75.0	yes
3	78.0	no
5	80.0	yes
10	80.0	no
14	80.0	yes
1	85.0	yes
2	90.0	yes
12	90.0	yes
8	95.0	yes
4	96.0	no

AVG = 76.5

ID	Humidity	Play
7	≤ 76.5	no
6	≤ 76.5	no
9	≤ 76.5	yes
11	≤ 76.5	yes
13	≤ 76.5	yes
3	> 76.5	no
5	> 76.5	yes
10	> 76.5	no
14	> 76.5	yes
1	> 76.5	yes
2	> 76.5	yes
12	> 76.5	yes
8	> 76.5	yes
4	> 76.5	no

NUMERIC ATTRIBUTE 5

- If humidity = 79.0 is used as the boundary, IG = 0.05.

ID	Humidity	Play
7	65.0	no
6	70.0	no
9	70.0	yes
11	70.0	yes
13	75.0	yes
3	78.0	no
5	80.0	yes
10	80.0	no
14	80.0	yes
1	85.0	yes
2	90.0	yes
12	90.0	yes
8	95.0	yes
4	96.0	no

AVG = 79.0

ID	Humidity	Play
7	≤ 79.0	no
6	≤ 79.0	no
9	≤ 79.0	yes
11	≤ 79.0	yes
13	≤ 79.0	yes
3	≤ 79.0	no
5	> 79.0	yes
10	> 79.0	no
14	> 79.0	yes
1	> 79.0	yes
2	> 79.0	yes
12	> 79.0	yes
8	> 79.0	yes
4	> 79.0	no

NUMERIC ATTRIBUTE 6

- If humidity = 82.5 is used as the boundary, IG = 0.05.

ID	Humidity	Play
7	65.0	no
6	70.0	no
9	70.0	yes
11	70.0	yes
13	75.0	yes
3	78.0	no
5	80.0	yes
10	80.0	no
14	80.0	yes
1	85.0	yes
2	90.0	yes
12	90.0	yes
8	95.0	yes
4	96.0	no

AVG = 82.5

ID	Humidity	Play
7	≤ 82.5	no
6	≤ 82.5	no
9	≤ 82.5	yes
11	≤ 82.5	yes
13	≤ 82.5	yes
3	≤ 82.5	no
5	≤ 82.5	yes
10	≤ 82.5	no
14	≤ 82.5	yes
1	> 82.5	yes
2	> 82.5	yes
12	> 82.5	yes
8	> 82.5	yes
4	> 82.5	no

NUMERIC ATTRIBUTE 7

- If humidity = 87.5 is used as the boundary, IG = 0.02.

ID	Humidity	Play
7	65.0	no
6	70.0	no
9	70.0	yes
11	70.0	yes
13	75.0	yes
3	78.0	no
5	80.0	yes
10	80.0	no
14	80.0	yes
1	85.0	yes
2	90.0	yes
12	90.0	yes
8	95.0	yes
4	96.0	no

AVG = 87.5

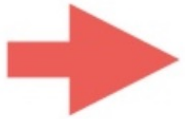
ID	Humidity	Play
7	≤ 87.5	no
6	≤ 87.5	no
9	≤ 87.5	yes
11	≤ 87.5	yes
13	≤ 87.5	yes
3	≤ 87.5	no
5	≤ 87.5	yes
10	≤ 87.5	no
14	≤ 87.5	yes
1	≤ 87.5	yes
2	> 87.5	yes
12	> 87.5	yes
8	> 87.5	yes
4	> 87.5	no

NUMERIC ATTRIBUTE 8

- If humidity = 92.5 is used as the boundary, IG = 0.01.

ID	Humidity	Play
7	65.0	no
6	70.0	no
9	70.0	yes
11	70.0	yes
13	75.0	yes
3	78.0	no
5	80.0	yes
10	80.0	no
14	80.0	yes
1	85.0	yes
2	90.0	yes
12	90.0	yes
8	95.0	yes
4	96.0	no

AVG = 92.5



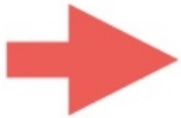
ID	Humidity	Play
7	≤ 92.5	no
6	≤ 92.5	no
9	≤ 92.5	yes
11	≤ 92.5	yes
13	≤ 92.5	yes
3	≤ 92.5	no
5	≤ 92.5	yes
10	≤ 92.5	no
14	≤ 92.5	yes
1	≤ 92.5	yes
2	≤ 92.5	yes
12	≤ 92.5	yes
8	> 92.5	yes
4	> 92.5	no

NUMERIC ATTRIBUTE 9

- If humidity = 95.5 is used as the boundary, IG = 0.01.

ID	Humidity	Play
7	65.0	no
6	70.0	no
9	70.0	yes
11	70.0	yes
13	75.0	yes
3	78.0	no
5	80.0	yes
10	80.0	no
14	80.0	yes
1	85.0	yes
2	90.0	yes
12	90.0	yes
8	95.0	yes
4	96.0	no

AVG = 95.5

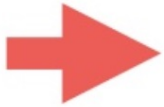


ID	Humidity	Play
7	≤ 95.5	no
6	≤ 95.5	no
9	≤ 95.5	yes
11	≤ 95.5	yes
13	≤ 95.5	yes
3	≤ 95.5	no
5	≤ 95.5	yes
10	≤ 95.5	no
14	≤ 95.5	yes
1	≤ 95.5	yes
2	≤ 95.5	yes
12	≤ 95.5	yes
8	> 95.5	yes
4	> 95.5	no

NUMERIC ATTRIBUTE 10

- The highest IG = 0.25 when the boundary is humidity = 72.5.

ID	Humidity	Play
7	65.0	no
6	70.0	no
9	70.0	yes
11	70.0	yes
13	75.0	yes
3	78.0	no
5	80.0	yes
10	80.0	no
14	80.0	yes
1	85.0	yes
2	90.0	yes
12	90.0	yes
8	95.0	yes
4	96.0	no



จุดตัด	IG
67.5	0.11
72.5	0.25
76.5	0.03
79.0	0.05
82.5	0.05
87.5	0.02
92.5	0.01
95.5	0.01

The highest IG

- Complex decision tree can become difficult for human to read.
- For people, you should simplify your model
  - ex: discretize your continuous variables in advance

n Information Gain (IG)

WHEN TO STOP GROWING A TREE?

- Complex tree can lead overfitting
- Two common critiria to stop splitting:
  - Set a minimum num. training inputs to use on each leaf
    - Ignore any leaf with less than n samples
  - Maximum depth
- Pruning to further increse the performance the tree
  - It involves removing the branches that make use of features having low importance.

ADVANTAGES AND DISADVANTAGES

ADVANTAGES OF CART

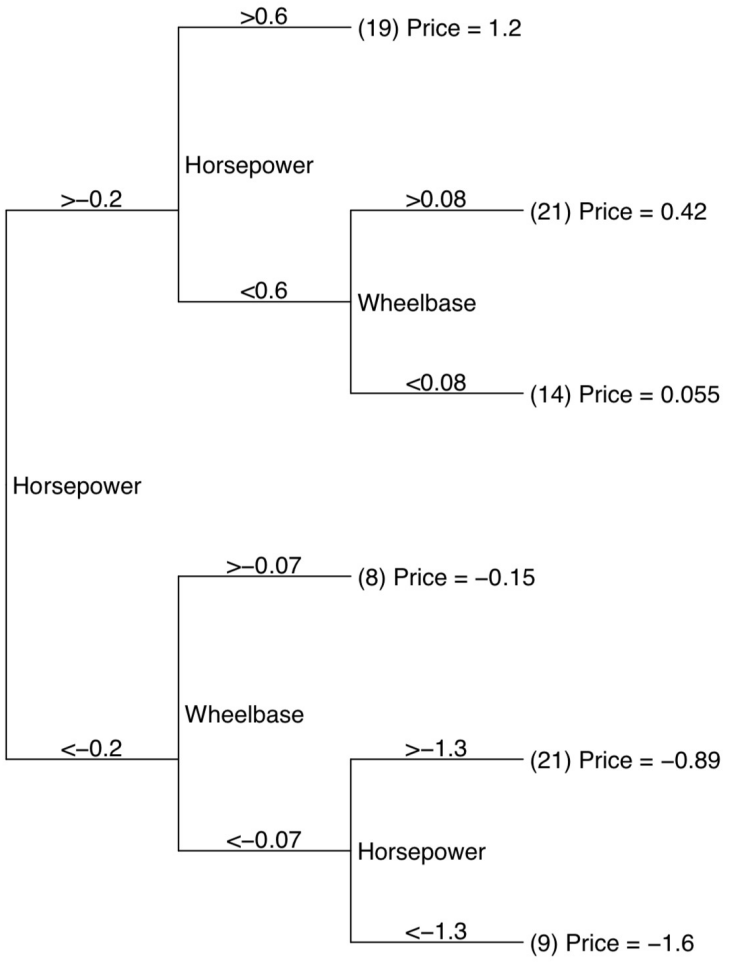
- Simple to understand, interpret, visualize.
- Implicitly perform variable screening or feature selection.
- Can handle both numerical and categorical data. Can also handle multi-output problems.
- Require relatively little effort for data preparation.
- Nonlinear relationships between parameters do not affect tree performance.

DISADVANTAGE OF CART

- Easy to have overfitting
- Can be unstable because small variations in the data might result in a completely different tree being generated. It needs to be lowered by methods like bagging and boosting.
- Greedy algorithms cannot guarantee to return the globally optimal decision tree.
- Create biased trees if some classes dominate.

REGRESSION TREE

- Regression tree for predicting price of cars.
- All features have been standardized to have zero mean and unit variance.
- Regression tree aims to minimize Mean Squared Error (MSE) rather than maximize IG at each node



PERFORMANCE MEASURES

CLASSIFICATION PROBLEM	REGRESSION PROBLEM
• Confusion matrix (Regression Tree Belongs here)	• MSE
• Precision and recall	• MAE
• F-measure	• MAPE
• Accuracy	• R^2
• ROC graph and area under curve (AUC)	

HW6: APPLY DECISION TREE TO PREDICT IMDB_SCORE (LOW OR HIGH) ON IMDB

- movie_title : Title of the Movie
- duration: Duration in minutes
- director_name : Name of the Director of the Movie.
- director_facebook_likes : Number of likes of the Director on his Facebook Page.
- color: Film colorization. 'Black and White' or 'Color'
- genres: Film categorization like 'Animation', 'Comedy', 'Romance', 'Horror', 'Sci-Fi', 'Action', 'Family'
- actor_1_name: Primary actor starring in the movie
- actor_1_facebook_likes : Number of likes of the Actor_1 on his/her Facebook Page.
- actor_2_name: Other actor starring in the movie
- actor_2_facebook_likes : Number of likes of the Actor_2 on his/her Facebook Page.
- actor_3_name: Other actor starring in the movie
- actor_3_facebook_likes : Number of likes of the Actor_3 on his/her Facebook Page.
- num_critic_for_reviews : Number of critical reviews on imdb
- num_voted_users: Number of people who voted for the movie
- cast_total_facebook_likes: Total number of facebook likes of the entire cast of the movie.
- language : English, Arabic, Chinese, French, German, Danish, Italian, Japanese etc
- country: Country where the movie is produced.
- gross: Gross earnings of the movie in Dollars
- budget: Budget of the movie in Dollars
- title_year: The year in which the movie is released (1916:2016)
- imdb_score: IMDB Score of the movie on IMDB
- movie_facebook_likes: Number of Facebook likes in the movie page.

HW6: SOLUTION STEPS 1

- Load data: Merge & Concatenation
- Explore data:
  - Head()/tail()
  - info()→data_type & size
  - describe() & value_counts() plot → distribution
  - isna()→missing value

HW6: SOLUTION STEPS 2

- Preprocess data: (DT requires little efforts for preprocessing. Do as option)
  - Split train-test
  - Encoding data features:
    - Integer → Float; ~~Categorical → Binary or Label Values~~
  - Fill in missing value in train dataset
  - ~~Standardize features (comparable units) in train dataset~~
  - Select features in train dataset
    - Only numerical features
    - ~~Correlation – use correlation matrix or heat map to ignore x which is highly correlated with another x~~

HW6: SOLUTION STEPS 3

- Model:
  - Debug the Decision Tree model to make the model work for train dataset
  - Preprocess (based on train dataset) and predict test dataset
  - Evaluate classification result of test dataset
  - Interpret the feature importance