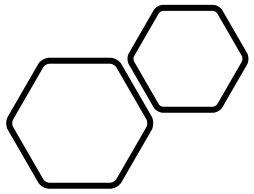
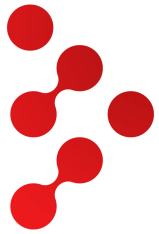




# Basic Machine Learning: Random Forest

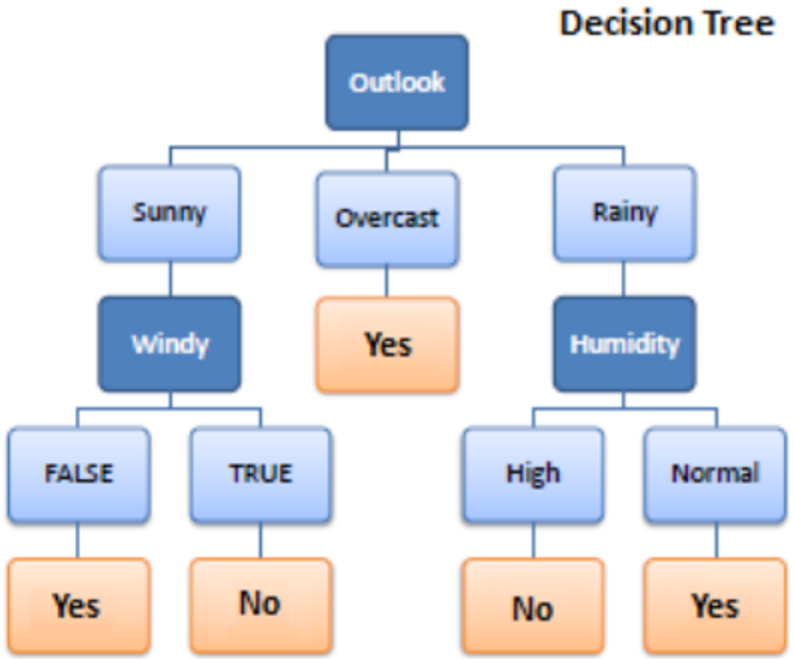
Oleh: Muhammad Angga Muttaqien | Founder & Mentor

# Content

# Decision Tree (Concept)

- Consider this example: What are the factors which decide if we going to play gold?
  - Outlook? (Sunny, Overcast, Rainy)
  - Temperature? (Hot, Mild, Cool)
  - Humidity? (High, Normal)
  - Windy? (False, True)

- Label: Play Golf? (Yes / No)



# Decision Tree (Concept)

| Predictors | | | | Target |
|---|---|---|---|---|
| Outlook | Temp. | Humidity | Windy | Play Golf |
| Rainy | Hot | High | False | No |
| Rainy | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Sunny | Mild | High | False | Yes |
| Sunny | Cool | Normal | False | Yes |
| Sunny | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Rainy | Mild | High | False | No |
| Rainy | Cool | Normal | False | Yes |
| Sunny | Mild | Normal | False | Yes |
| Rainy | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Sunny | Mild | High | True | No |

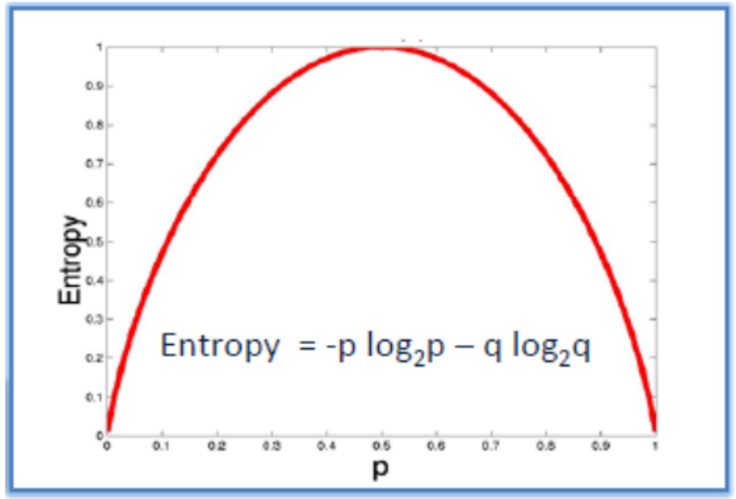
# Decision Tree (Concept)

Find the entropy the target feature.

a. If sample is completely homogenous, entropy = 0
b. If sample is equally divided, entropy = 1

| Play Golf | |
|---|---|
| Yes | No |
| 9 | 5 |

**Entropy(PlayGolf)** = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94




Entropy = -0.5 $\log_2$0.5 – 0.5 $\log_2$0.5 = 1

# Decision Tree (Concept)

Find the entropy of each feature towards the target.

| | | Play Golf | | |
|---|---|---|---|---|
| | | Yes | No | |
| Outlook | Sunny | 3 | 2 | 5 |
| | Overcast | 4 | 0 | 4 |
| | Rainy | 2 | 3 | 5 |
| | | | | 14 |

**E**(PlayGolf, Outlook) = **P**(Sunny)*__E__(3,2) + **P**(Overcast)*__E__(4,0) + **P**(Rainy)*__E__(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

# Decision Tree (Concept)

- Find the information gain of each feature used to predict the target.
- Information gain: Decrease of entropy after the dataset is split on an attribute
  - Creating decision tree classification is about finding attribute that return the highest IG

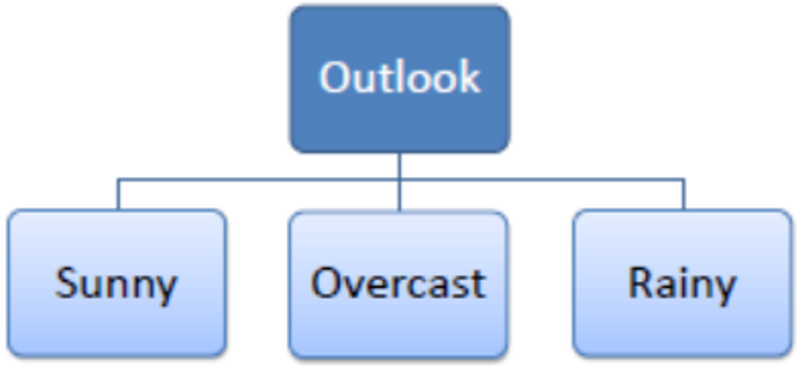
$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

**G**(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

# Decision Tree (Concept)

Select feature with the highest Information Gain as the root node.



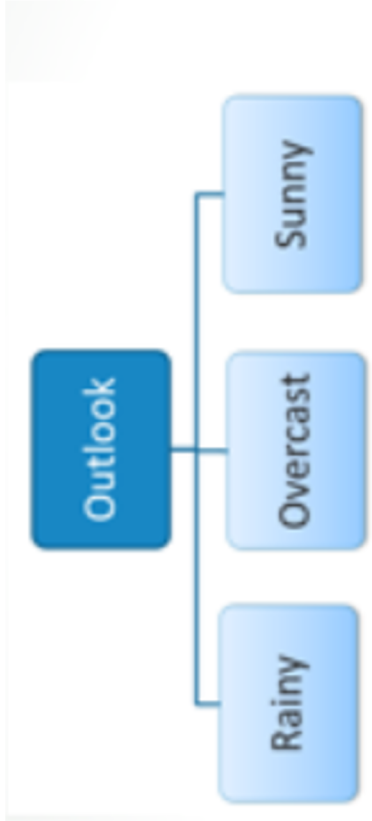

| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Outlook | Sunny | 3 | 2 |
| | Overcast | 4 | 0 |
| | Rainy | 2 | 3 |
| Gain = 0.247 | | | |

| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Temp. | Hot | 2 | 2 |
| | Mild | 4 | 2 |
| | Cool | 3 | 1 |
| Gain = 0.029 | | | |

| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Humidity | High | 3 | 4 |
| | Normal | 6 | 1 |
| Gain = 0.152 | | | |

| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Windy | False | 6 | 2 |
| | True | 3 | 3 |
| Gain = 0.048 | | | |

# Decision Tree (Concept)

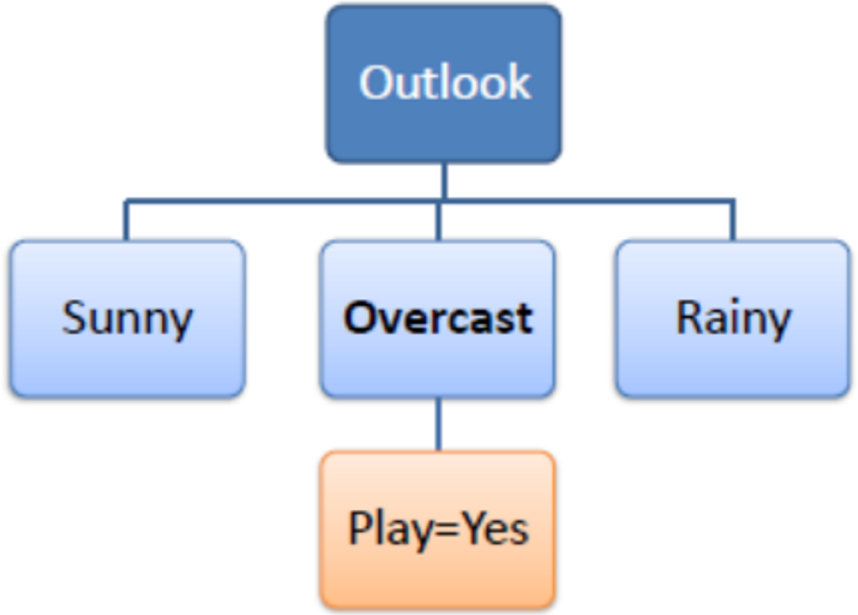



| Outlook | Temp | Humidity | Windy | Play Golf |
|---|---|---|---|---|
| Sunny | Mild | High | FALSE | Yes |
| Sunny | Cool | Normal | FALSE | Yes |
| Sunny | Cool | Normal | TRUE | No |
| Sunny | Mild | Normal | FALSE | Yes |
| Sunny | Mild | High | TRUE | No |
| Overcast | Hot | High | FALSE | Yes |
| Overcast | Cool | Normal | TRUE | Yes |
| Overcast | Mild | High | TRUE | Yes |
| Overcast | Hot | Normal | FALSE | Yes |
| Rainy | Hot | High | FALSE | No |
| Rainy | Hot | High | TRUE | No |
| Rainy | Mild | High | FALSE | No |
| Rainy | Cool | Normal | FALSE | Yes |
| Rainy | Mild | Normal | TRUE | Yes |

# Decision Tree (Concept)

Branch with entropy = 0 is a terminal node (leaf)

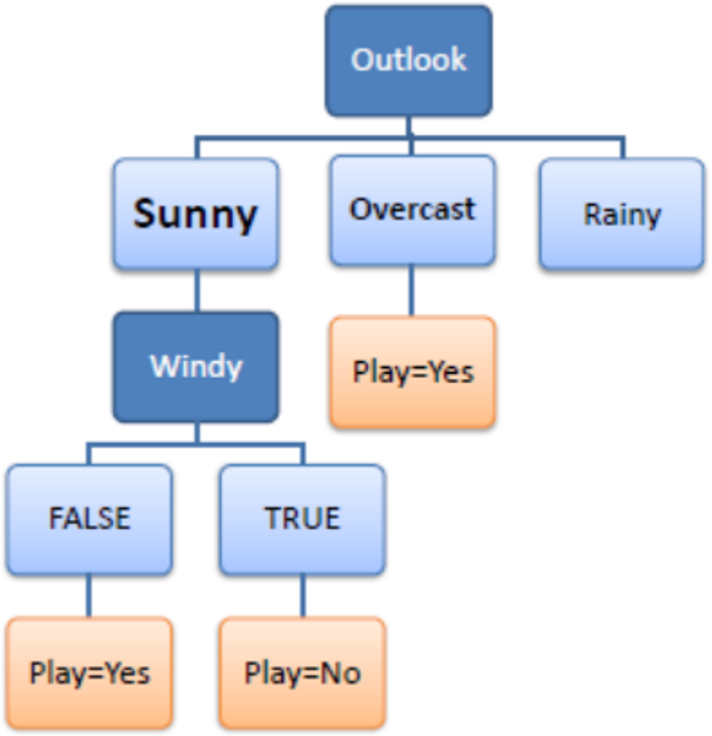
| Temp. | Humidity | Windy | Play Golf |
|---|---|---|---|
| Hot | High | FALSE | Yes |
| Cool | Normal | TRUE | Yes |
| Mild | High | TRUE | Yes |
| Hot | Normal | FALSE | Yes |



# Decision Tree (Concept)

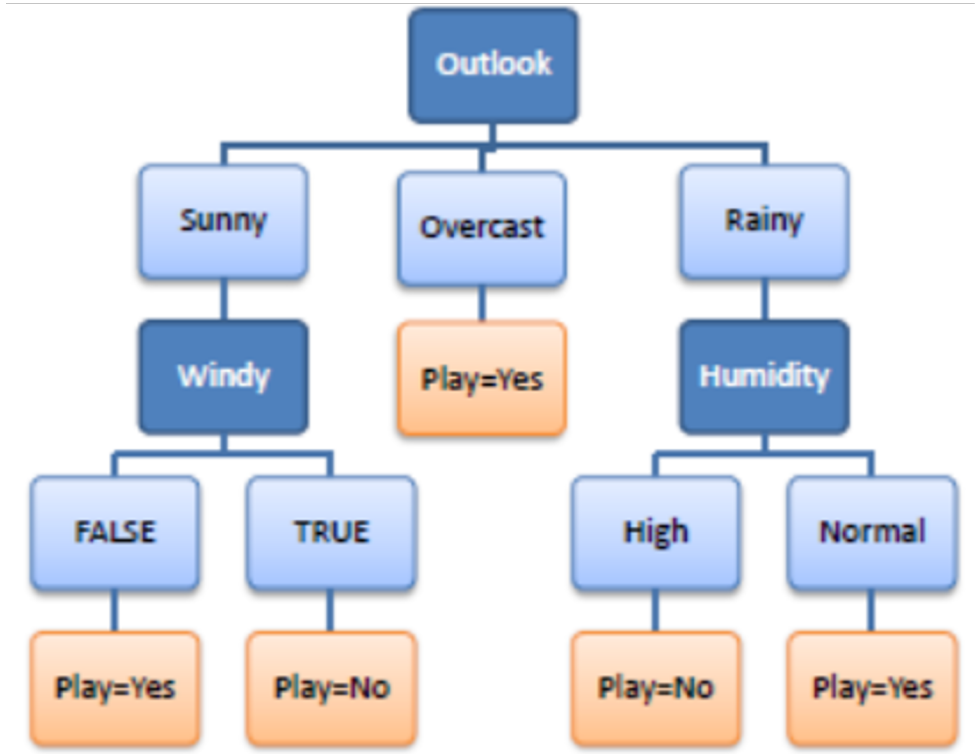
Branch with entropy > 0 needs further splitting

| Temp | Humidity | Windy | Play Golf |
|---|---|---|---|
| Mild | High | FALSE | Yes |
| Cool | Normal | FALSE | Yes |
| Mild | Normal | FALSE | Yes |
| Cool | Normal | TRUE | No |
| Mild | High | TRUE | No |



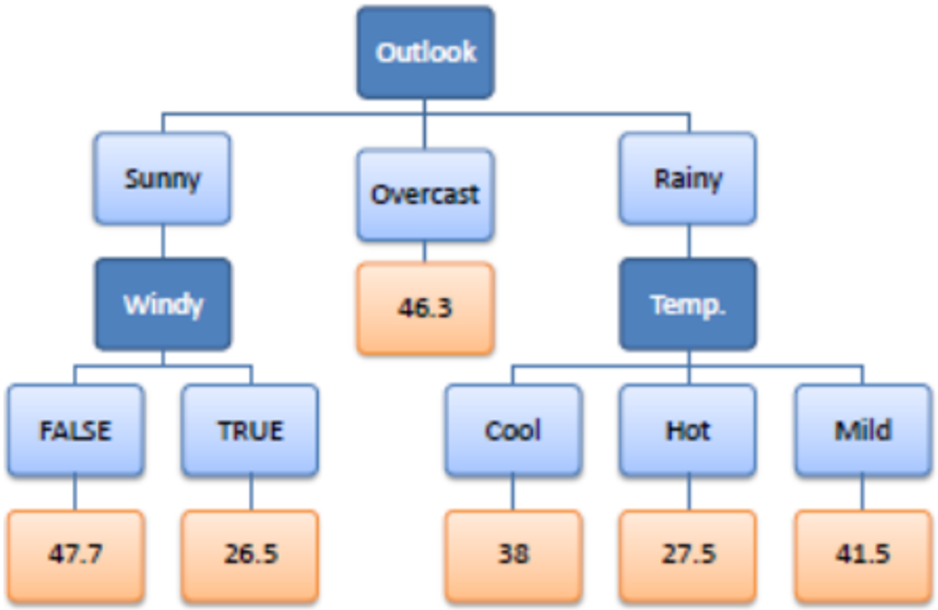
# Decision Tree (Concept)

Decision tree algorithm is run recursively on the non-leaf branches, until all data is classified



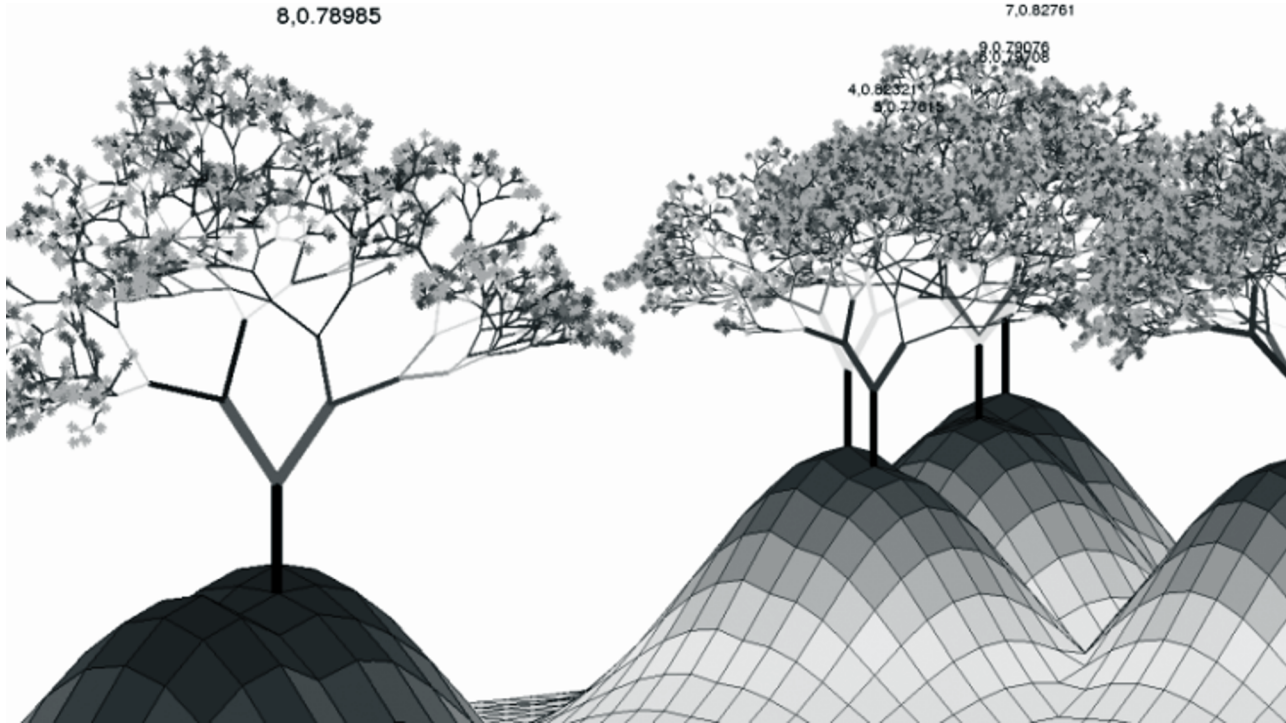
# Decision Tree (Concept)

| Predictors | | | | Target |
|---|---|---|---|---|
| Outlook | Temp. | Humidity | Windy | Hours Played |
| Rainy | Hot | High | False | 25 |
| Rainy | Hot | High | True | 30 |
| Overcast | Hot | High | False | 48 |
| Sunny | Mild | High | False | 45 |
| Sunny | Cool | Normal | False | 52 |
| Sunny | Cool | Normal | True | 23 |
| Overcast | Cool | Normal | True | 43 |
| Rainy | Mild | High | False | 35 |
| Rainy | Cool | Normal | False | 38 |
| Sunny | Mild | Normal | False | 46 |
| Rainy | Mild | Normal | True | 48 |
| Overcast | Mild | High | True | 52 |
| Overcast | Hot | Normal | False | 44 |
| Sunny | Mild | High | True | 30 |



# Random Forest (Concept)

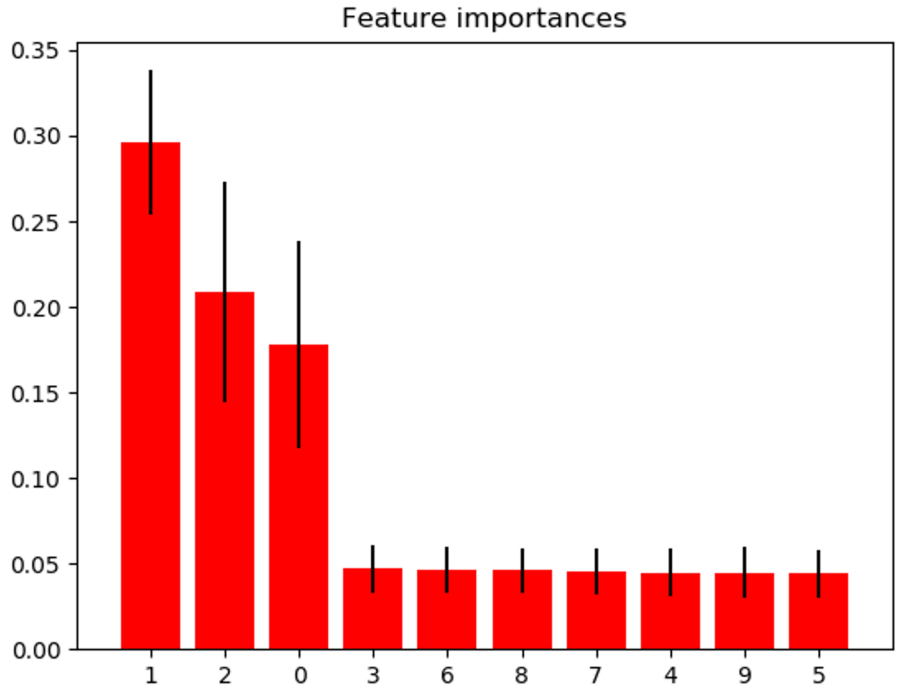
- Random forest grow multiple trees as opposed to a single CART (Classification And Regression Tree)
  - The classification forest chooses the result with  most votes (over all trees)
  - The regression forest average outputs of different trees
- Random forest use 2 method:
  - Bagging (Bootstrap Aggregating)
  - Random subspace method.



# Random Forest (Concept)

Shows the use of forests of trees to evaluate the importance of features on an artificial classification task. The red bars are the feature importances of the forest, along with their inter-trees variability.

As expected, the plot suggests that 3 features are informative, while the remaining are not.



# Random Forest (Concept)

Features used at the top of the tree contribute to the final prediction

decision of a larger fraction of the input samples

```
Feature ranking:
1. feature 1 (0.295902)
2. feature 2 (0.208351)
3. feature 0 (0.177632)
4. feature 3 (0.047121)
5. feature 6 (0.046303)
6. feature 8 (0.046013)
7. feature 7 (0.045575)
8. feature 4 (0.044614)
9. feature 9 (0.044577)
10. feature 5 (0.043912)
```

# Thanks!