AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

UNIVERSITY OF SOUTH CAROLINA

CSCE 580: Introduction to AI
CSCE 581: Trusted AI

Lecture 13: Machine Learning

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

5TH OCT 2023

Carolinian Creed: "I will practice personal and academic integrity."

Organization of Lecture 13

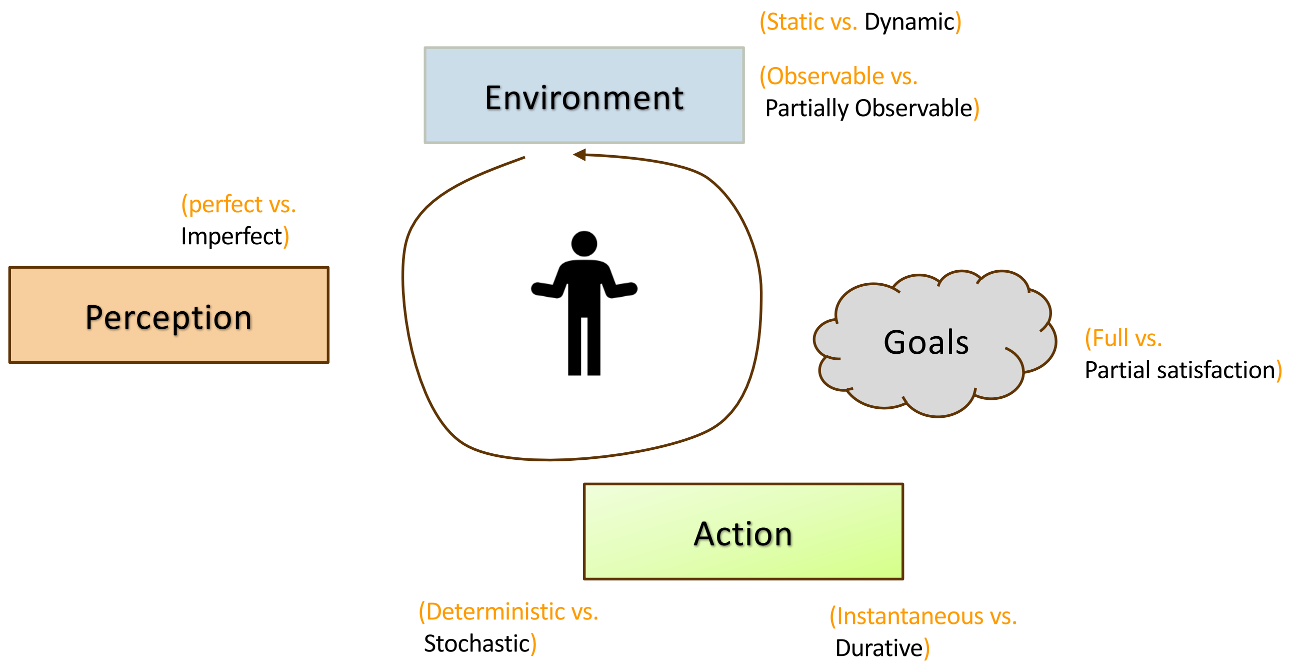
- Introduction Segment
  - Recap of Lecture 12
- Main Segment
  - The variety of methods fpr classification
  - Logistic Regression
  - Decision trees
  - Random forest
  - Discussion: Choosing a method that works
- Concluding Segment
  - Course Project Discussion
  - About Next Lecture – Lecture 14
  - Ask me anything

Introduction Section

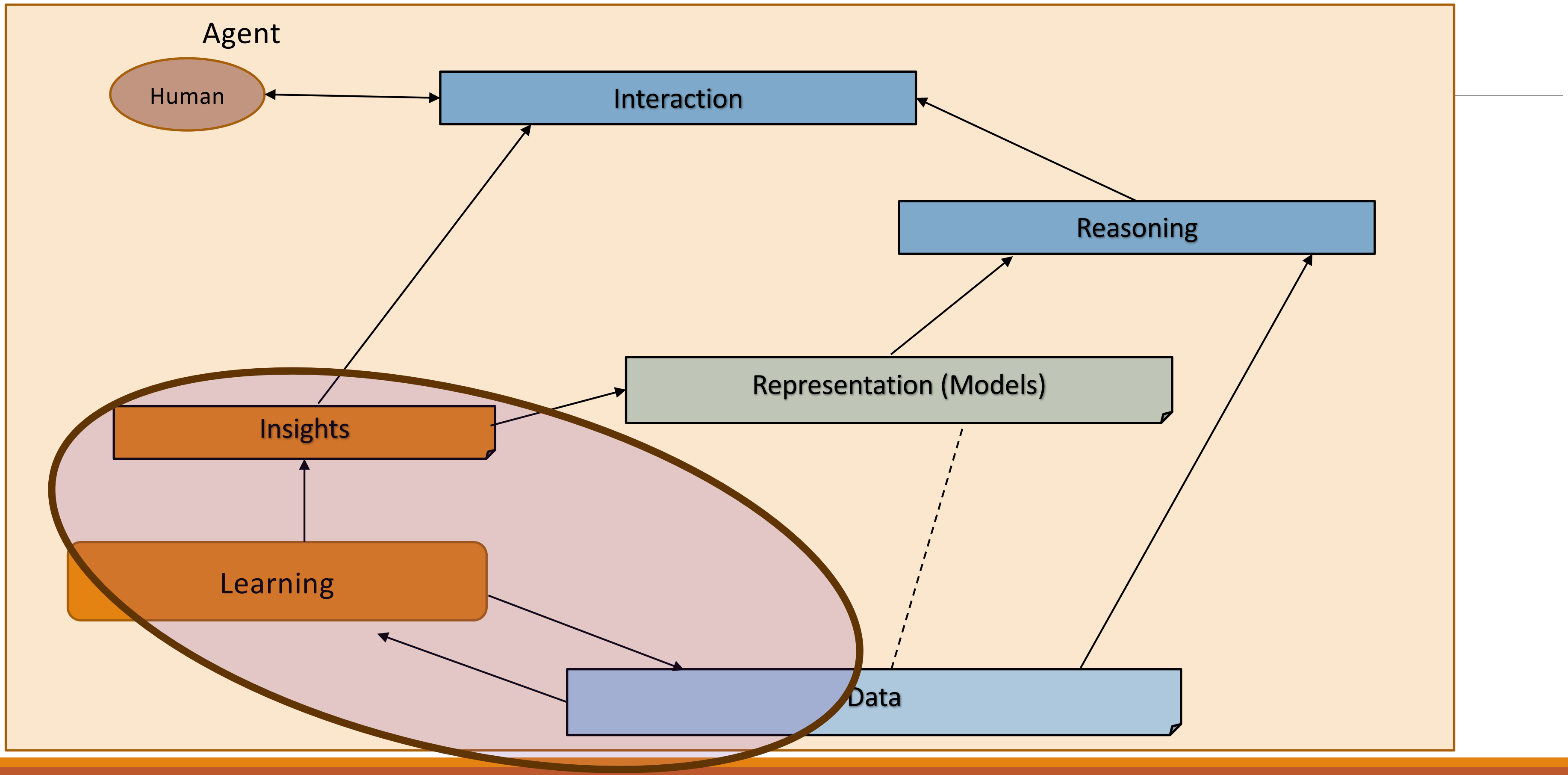
Recap of Lecture 12

- Problem Settings
- Data preparation and feature engineering
- Solving classification problems

Intelligent Agent Model


(Static vs. Dynamic)
Environment
(Observable vs. Partially Observable)
(perfect vs. Imperfect)
Perception
Goals
(Full vs. Partial satisfaction)
Action
(Deterministic vs. Stochastic)
(Instantaneous vs. Durative)

Relationship Between Main AI Topics



Where We Are in the Course

CSCE 580/ 581 – In This Course

- Week 1: Introduction, Aim: Chatbot / Intelligence Agent
- Weeks 2-3: Data: Formats, Representation and the Trust Problem
- Week 4-5: Search, Heuristics - Decision Making
- Week 6: Constraints, Optimization – Decision Making
- Week 7: Classical Machine Learning – Decision Making, Explanation
- Week 8: Machine Learning - Classification
- Week 9: Machine Learning - Classification – Trust Issues and Mitigation Methods
- Topic 10: Learning neural network, deep learning, Adversarial attacks
- Week 11: Large Language Models – Representation, Issues
- Topic 12: Markov Decision Processes, Hidden Markov models - Decision making
- Topic 13: Planning, Reinforcement Learning – Sequential decision making
- Week 14: AI for Real World: Tools, Emerging Standards and Laws; Safe AI/ Chatbots

Main Section

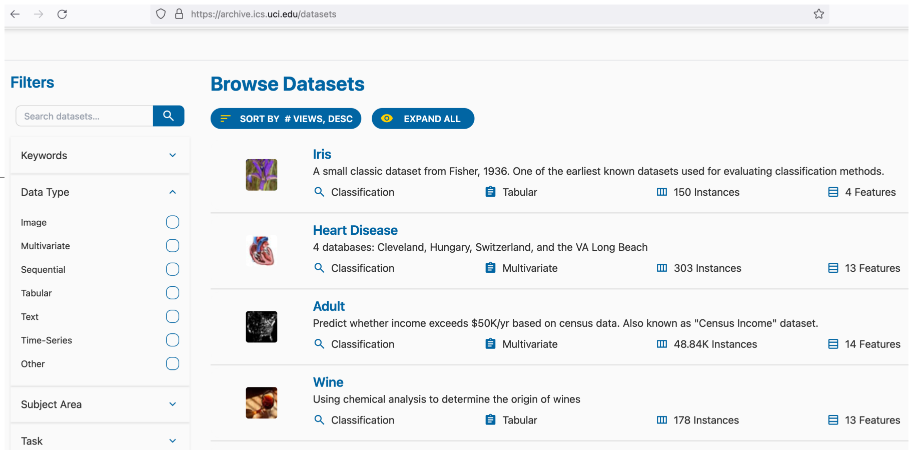
Credit: Retrieved from internet

Machine Learning – Insights from Data

- Descriptive analysis
  - Describe a past phenomenon
  - **Methods**: classification (feedback from label), clustering, dimensionality reduction, anomaly detection, neural methods, reinforcement learning (feedback from hint/ reward)
- Predictive analysis
  - Predict about a new situation
  - **Methods**: time-series, neural networks
- Prescriptive analysis
  - What an agent should do
  - **Methods**: simulation, reinforcement learning, reasoning
- New areas
  - Counterfactual analysis
  - Causal Inferencing
  - Scenario planning

Reference and Demo

- Data: UCI Datasets
  - https://archive.ics.uci.edu/datasets
  - Browse or search


https://archive.ics.uci.edu/datasets

Filters

Search datasets...

Keywords

Data Type

Image ◯
Multivariate ◯
Sequential ◯
Tabular ◯
Text ◯
Time-Series ◯
Other ◯

Subject Area

Task

Browse Datasets

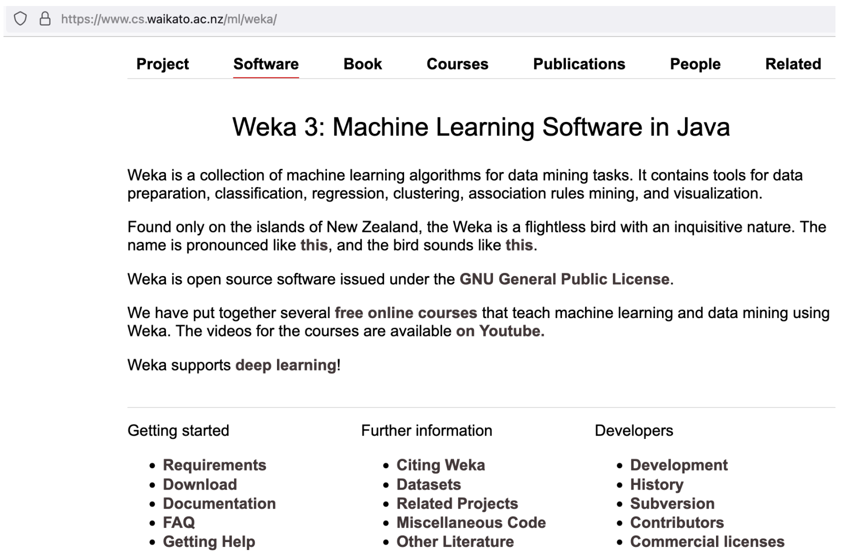
SORT BY # VIEWS, DESC | EXPAND ALL

Iris
A small classic dataset from Fisher, 1936. One of the earliest known datasets used for evaluating classification methods.
Classification | Tabular | 150 Instances | 4 Features

Heart Disease
4 databases: Cleveland, Hungary, Switzerland, and the VA Long Beach
Classification | Multivariate | 303 Instances | 13 Features

Adult
Predict whether income exceeds $50K/yr based on census data. Also known as "Census Income" dataset.
Classification | Multivariate | 48.84K Instances | 14 Features

Wine
Using chemical analysis to determine the origin of wines
Classification | Tabular | 178 Instances | 13 Features


https://www.cs.waikato.ac.nz/ml/weka/

Project | Software | Book | Courses | Publications | People | Related

Weka 3: Machine Learning Software in Java

Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization.

Found only on the islands of New Zealand, the Weka is a flightless bird with an inquisitive nature. The name is pronounced like **this**, and the bird sounds like **this**.

Weka is open source software issued under the **GNU General Public License**.

We have put together several **free online courses** that teach machine learning and data mining using Weka. The videos for the courses are available **on Youtube**.

Weka supports **deep learning**!

Getting started
- **Requirements**
- **Download**
- **Documentation**
- **FAQ**
- **Getting Help**

Further information
- **Citing Weka**
- **Datasets**
- **Related Projects**
- **Miscellaneous Code**
- **Other Literature**

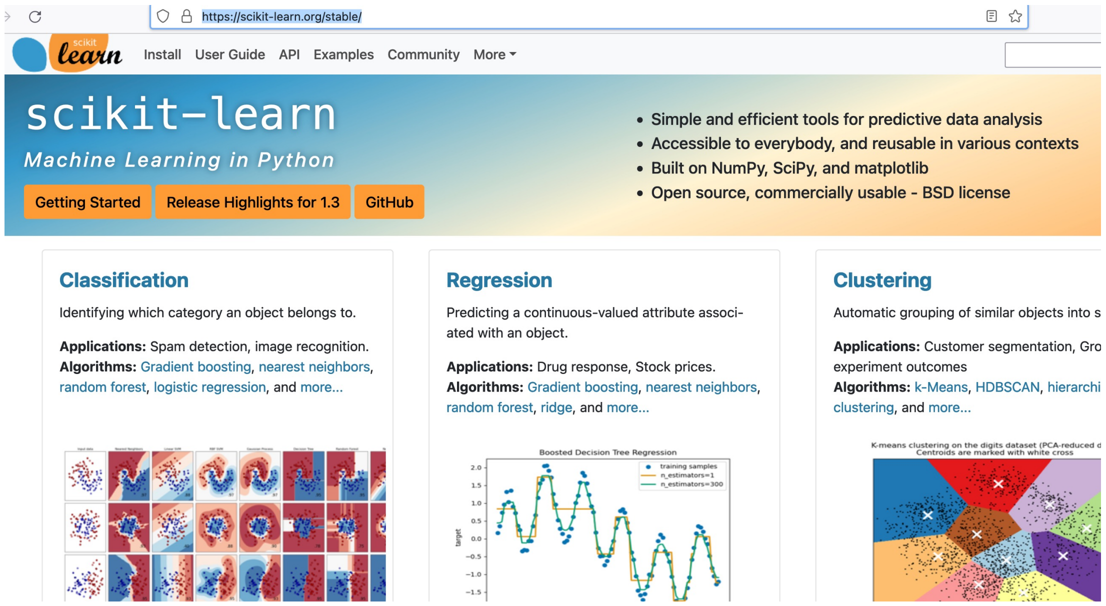
Developers
- **Development**
- **History**
- **Subversion**
- **Contributors**
- **Commercial licenses**

- Tools:
  - Weka - https://www.cs.waikato.ac.nz/ml/weka/
  - Download tool and dataset

- Libraries
  - Scikit - https://scikit-learn.org/stable/

Reference and Demo

- Data: UCI Datasets
  - https://archive.ics.uci.edu/datasets
  - Browse or search
- Tools:
  - Weka - https://www.cs.waikato.ac.nz/ml/weka/
  - Download tool and dataset
- Libraries
  - Scikit - https://scikit-learn.org/stable/



scikit-learn

Machine Learning in Python

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

Regression

Predicting a continuous-valued attribute associ-ated with an object.

Applications: Drug response, Stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

Clustering

Automatic grouping of similar objects into s

Applications: Customer segmentation, Gro experiment outcomes
Algorithms: k-Means, HDBSCAN, hierarchi clustering, and more...

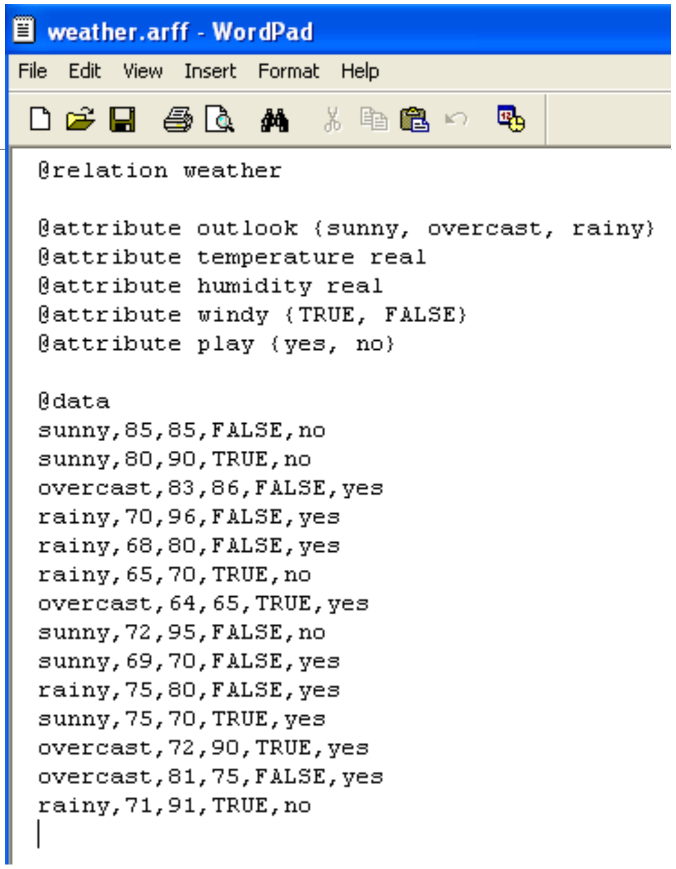
Exercise: Run Weka

Exercise: Understand German Credit

- Check in UCI
- Look at variants
  - https://archive.ics.uci.edu/dataset/573/south+german+credit+update

ARFF Data Format

- Attribute-Relation File Format
- Header – describing the attribute types
- Data – (instances, examples) comma-separated list



```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

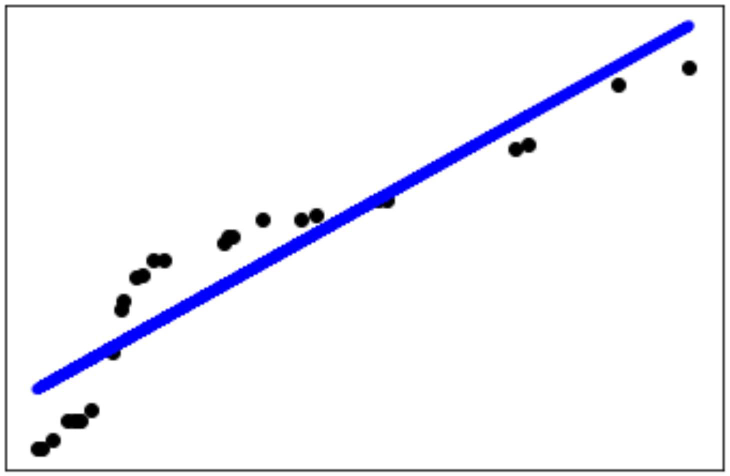
Slide Courtesy: http://www.cs.iastate.edu/~cs573x/bbsilab.html

Classifier Method Types

- Individual methods
  - Logistic Regression
  - Decision Tree
  - Naïve Bayes
- Ensemble
  - Bagging: Aggregate classifiers (“bootstrap aggregation” => bagging)
    - Random Forest
    - Samples are chosen with replacement (bootstrapping), and combined (aggregated) by taking their average
  - Gradient Boosting: aggregate to turn weak learners into strong learners
    - Boosters (aggregators) turn weak learners into strong learners by focusing on where the individual weak models (decision trees, linear regressors) went wrong
    - Gradient Boosting

Source:
- Data Mining: Concepts and Techniques, by Jiawei Han and Micheline Kamber
- https://towardsdatascience.com/getting-started-with-xgboost-in-scikit-learn-f69f5f470a97

Linear Regression



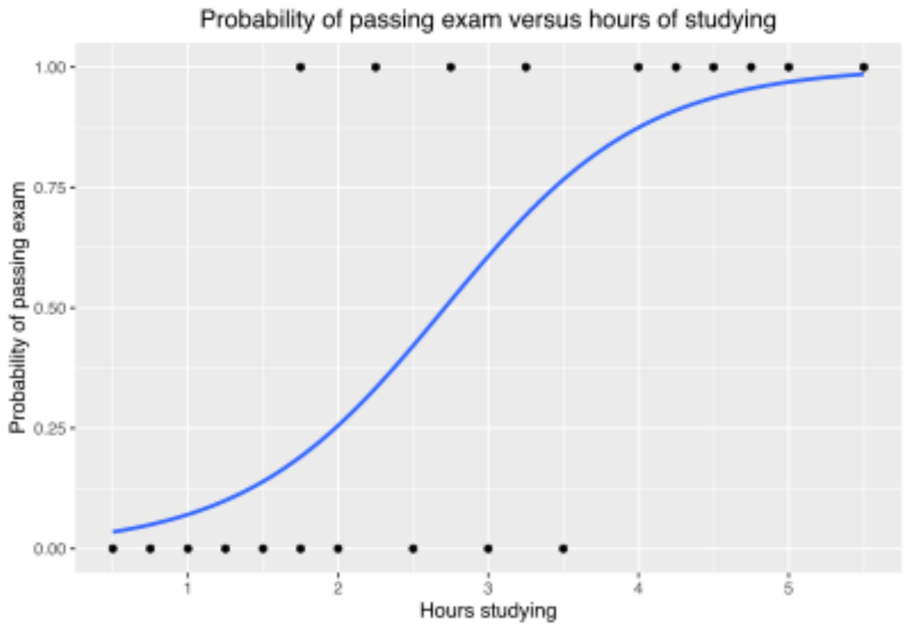
Notebook: https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l6-l7-l8-supervised-ml/Supervised-Regression.ipynb

Logistic Regression

- Classification setting – labels are discrete
  - the **logistic model** (or **logit model**) is a statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables

$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}}$$

where μ is a location parameter (the midpoint of the curve) and s is a scale parameter.



$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Source: Wikipedia

•Sources:
- •https://www.ibm.com/topics/logistic-regression
- •https://developers.google.com/machine-learning/crash-course/logistic-regression/
- •https://en.wikipedia.org/wiki/Logistic_regression
- •https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Decision Tree

Problem: Classify Weather Data

Input

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
…	…	…	…	…

Class Label

Output (Informal)

```
If outlook = sunny and humidity = high then play = no
If outlook = rainy and windy = true then play = no
If outlook = overcast then play = yes
If humidity = normal then play = yes
If none of the above then play = yes
```

Slide Adapted From/ Courtesy: Data Mining, Practical Machine Learning Tools and Techniques, Slides for Chapter 6, Trees and Rules of Data Mining by I. H. Witten, E. Frank, M. A. Hall and C. J. Pal

Which Variable to Learn to Create Rules On?

- What do we want?
  - Compact model (e.g., set of rules)
  - High accuracy / low error
- Find the most discriminating variable
  - But how do we measure this
- Corner cases
  - If all the samples are the same, the decision tree is a ?
    - Leaf node with the only class
  - If there are no attributes in the dataset, the decision tree is?
    - A node with most common class

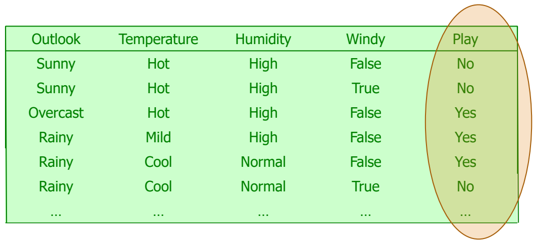
Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
…	…	…	…	…

Expected Information/ Entropy

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
…	…	…	…	…

- Concept: Expected Information
  - Let
    - Class label has **m** distinct values (i.e., m distinct classes)
    - s_i be the number of samples of S of Class C_i (i = 1 ..m)
  - $I(s_1, s_2, .., s_m) = - \Sigma_{i=1 \text{ to } m}\, p_i \log_2 (p_i)$
    - Where P_i is the probability a sample belongs to class Ci; estimated by(s_i /s)

- Entropy / Expected Information after partitioning on Attribute A which has **v** distinct values
  - $E(A) = \Sigma_{j=1 \text{ to } v}\ (s_{1j} + \ldots + s_{mj}) / S \quad * (I(s_{1j}, s_{2j}, .., s_{mj}))$
  - s_{ij} be the number of samples in S_j of Class C_i (i = 1 ..m)
  - Smaller the entropy, the greater the purity of the subset partitions



Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
…	…	…	…	…

Information Gain

- Entropy / Expected Information after partitioning on Attribute A which has v distinct values
  - $E(A) = \Sigma_{j=1 \text{ to } v} \; (s_{1j} + \ldots + s_{mj}) / S \;\; * (I(s_{1j}, s_{2j}, .., s_{mj})$
    - s_{ij} be the number of samples in S_j of Class C_i ($i = 1 .. m$)

- After partition, S_j
  - $I(s_{1j}, s_{2j}, .., s_{mj}) = - \Sigma_{i=1 \text{ to } m} \, p_{ij} \log_2 (p_{ij})$
  - Where p_{ij} is the probability a sample in S_j belongs to class C_i; estimated by(s_{ij} / | s_j |)

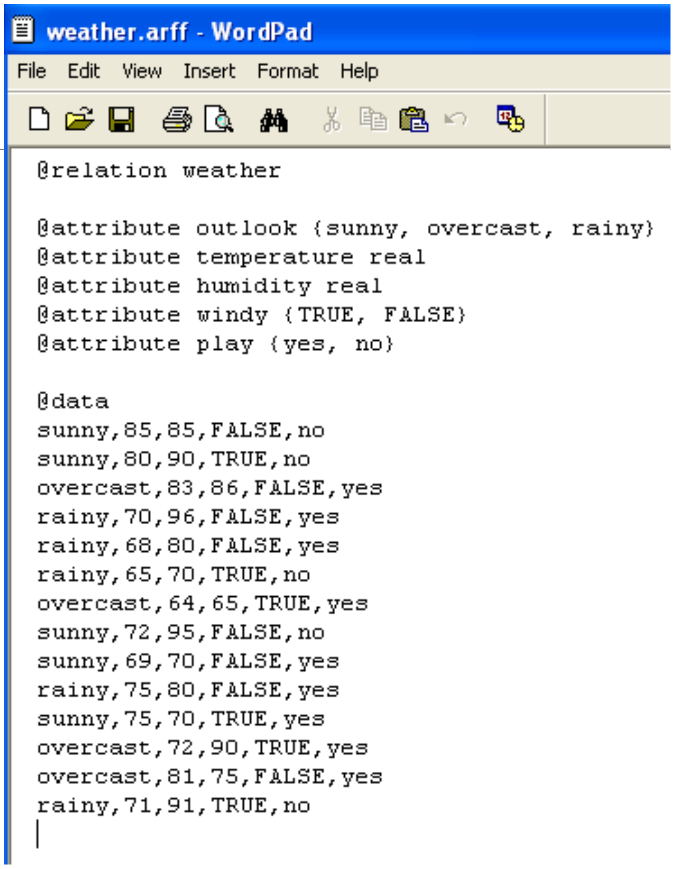
- Gain (A) = $I(s_1, s_2, .., s_m)$ - E(A)
  - Is the expected reduction in entropy by knowing the value of Attribute A

- **Method**: Split on the attribute which leads to the highest information gain

Weka Exercise

ARFF Data Format

- Data is in ARFF in UCI dataset
- Or Convert
  - File system, CSV → ARFF format
  - Use [C45Loader]() and [CSVLoader]() to convert

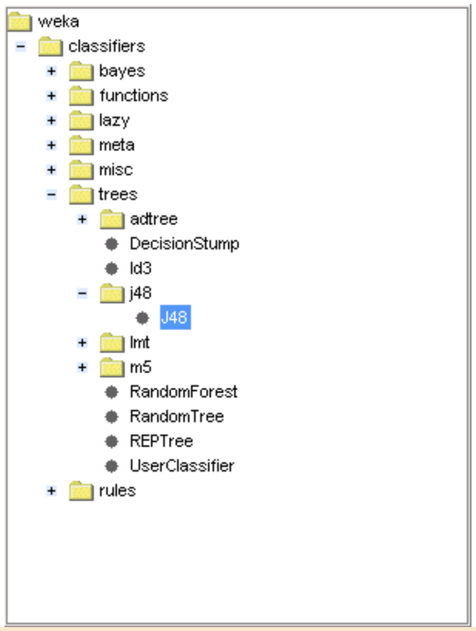


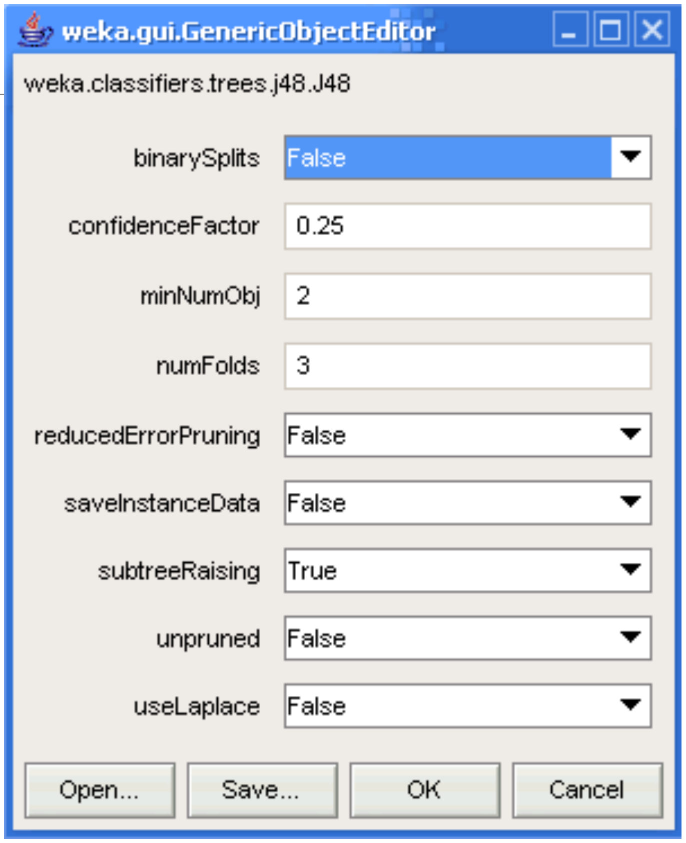
```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

Slide Courtesy: http://www.cs.iastate.edu/~cs573x/bbsilab.html

Weka: weka.classifiers.trees.J48

Class for generating an unpruned or a pruned C4.5 decision tree.



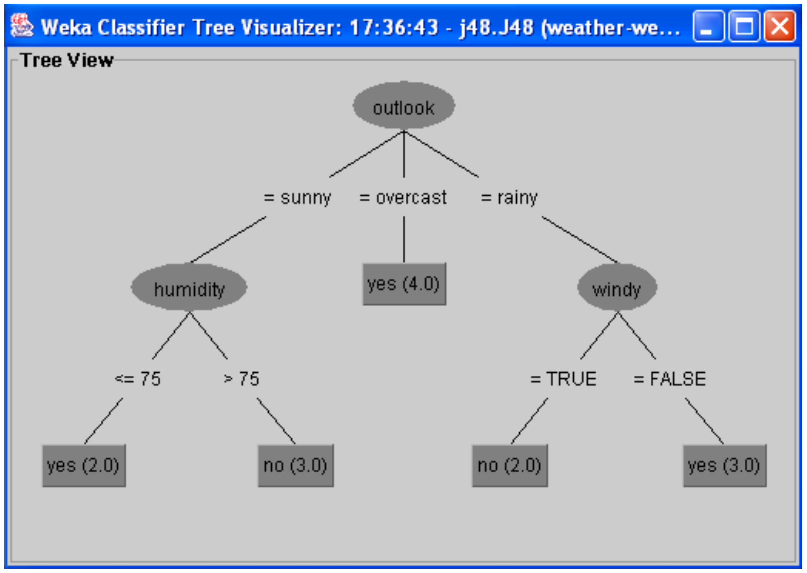


weka.classifiers.trees.j48.J48

binarySplits	False
confidenceFactor	0.25
minNumObj	2
numFolds	3
reducedErrorPruning	False
saveInstanceData	False
subtreeRaising	True
unpruned	False
useLaplace	False

Slide Courtesy: http://www.cs.iastate.edu/~cs573x/bbsilab.html

Understanding Output



Slide Courtesy: http://www.cs.iastate.edu/~cs573x/bbsilab.html

Weka: Decision Tree Output

```
J48 pruned tree
------------------


outlook = sunny
|   humidity = high: no (3.0)
|   humidity = normal: yes (2.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = TRUE: no (2.0)
|   windy = FALSE: yes (3.0)

Number of Leaves  :     5

Size of the tree :      8
```

```
=== Summary ===

Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                         -0.0426
Mean absolute error                      0.4167
Root mean squared error                  0.5984
Relative absolute error                 87.5    %
Root relative squared error            121.2987 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.556     0.6       0.625     0.556     0.588      0.633    yes
                 0.4       0.444     0.333     0.4       0.364      0.633    no
Weighted Avg.    0.5       0.544     0.521     0.5       0.508      0.633

=== Confusion Matrix ===

 a b   <-- classified as
 5 4 | a = yes
 3 2 | b = no
```

Slide Courtesy: http://www.cs.iastate.edu/~cs573x/bbsilab.html

Test Options

- Percentage Split (2/3 Training; 1/3 Testing)
- Cross-validation
  - Estimating the generalization error based on resampling when limited data
    - averaged error estimate.
  - Cross-fold validation (10-fold)
  - Leave-one-out (Loo)
  - Stratified

Slide Courtesy: http://www.cs.iastate.edu/~cs573x/bbsilab.html

Random Forest

- An ensemble method
- Credits
  - Ideas introduced by Tin Kam Ho in 1995, https://en.wikipedia.org/wiki/Tin_Kam_Ho
  - Matured by Leo Breiman and Adele Cutler at Berkeley (https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#intro)
  - History: Khaled Fawagreh, Mohamed Medhat Gaber & Eyad Elyan (2014) Random forests: from early developments to recent advancements, Systems Science & Control Engineering, 2:1, 602-609, DOI: 10.1080/21642583.2014.956265
- Main steps (Input: data, N= number of trees)
  - If the number of cases in the training set is N, sample N cases at random - but *with replacement*, from the original data. This sample will be the training set for growing the tree.
  - If there are M input variables, a number m<<M is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
  - Each tree is grown to the largest extent possible. There is no pruning.

Slide Courtesy: Leo Breiman and Adele Cutler website

Random Forest in Action

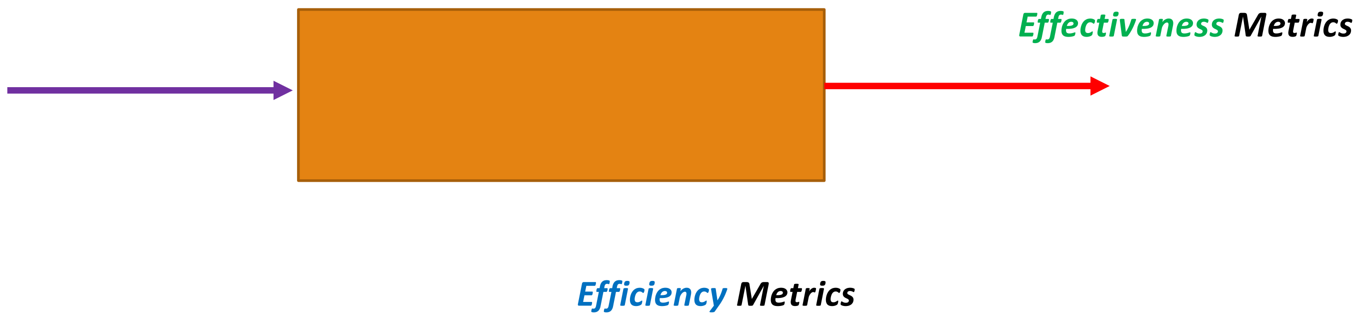
- Code examples:
  - https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l6-l7-supervised-ml/Supervised-RandomForest-Classification.ipynb
- Scikit Library: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Activity: Try Weka and Classifiers

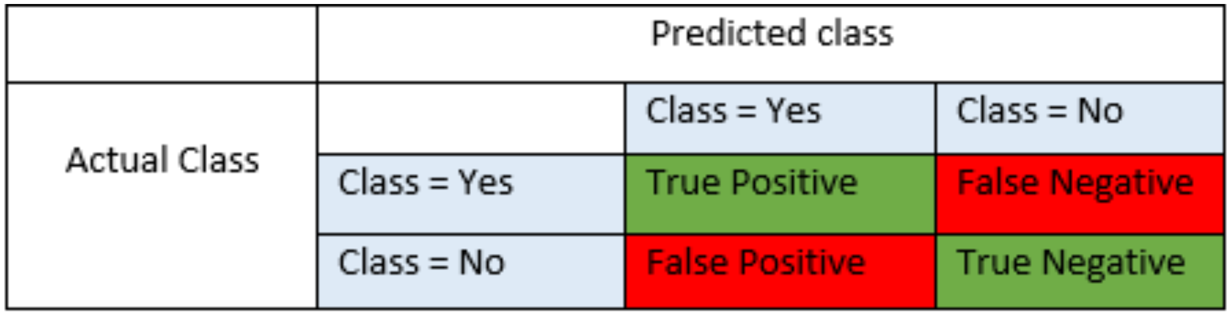
- Naïve Bayes Method
- Gradient Tree Boosting
- Neural Network – MLP

Metric Types

- **Effectiveness**: what the **user** of a system sees, primarily cares about
- **Efficiency**: what the **executor** in a system sees, primarily cares about



Metrics: Accuracy, Precision, Recall



	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Accuracy =
(TP+TN)/
(TP+FP+FN+TN)

Precision =
(TP)/
(TP+FP)

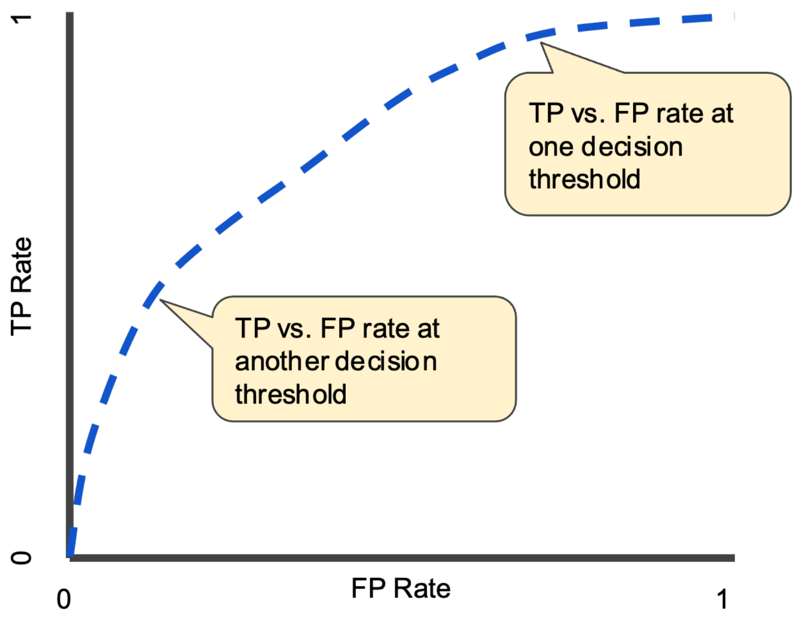
Recall =
(TP)/
(TP+FN)

F1 Score: *Harmonic Mean*

1/F1 = 1/Precision + 1/Recall

F1 = 2*(Recall * Precision) /
(Recall + Precision)

ROC – Receiver Operating Characteristic curve



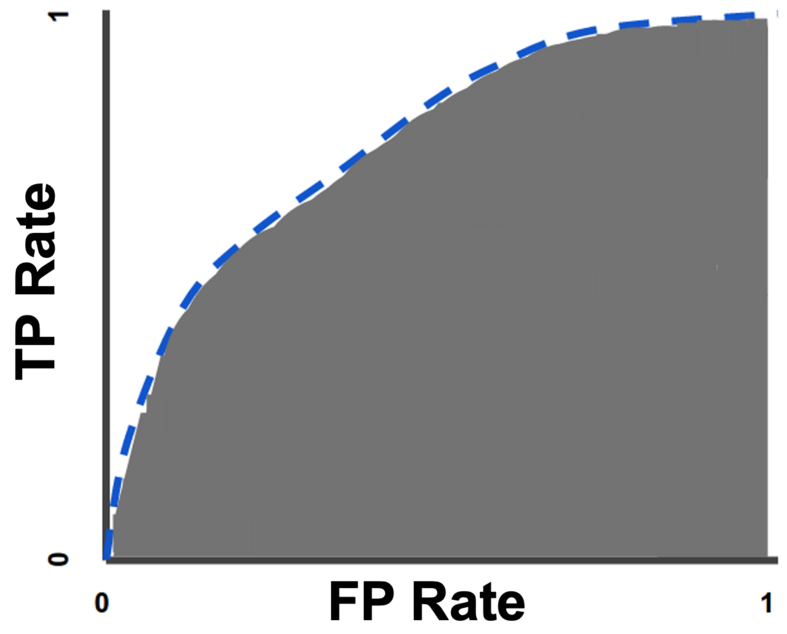
True Positive Rate = Recall =
(TP)/
(TP+FN)

False Positive Rate =
(FP)/
(FP+TN)

		Predicted class	
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Source: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

AUC – Area Under the ROC Curve



- Aggregate measure of performance across all possible classification thresholds.
- Interpretation: probability that the model ranks a random positive example more highly than a random negative example

Source: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

References

- Blogs: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/
- Google: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Discussion: 10 Tips Paper

- Access: https://biodatamining.biomedcentral.com/articles/10.1186/s13040-017-0155-3
- Chicco, D. Ten quick tips for machine learning in computational biology. *BioData Mining* **10,** 35 (2017). https://doi.org/10.1186/s13040-017-0155-3

The Tips

- Tip 1: Check and arrange your input dataset properly
- Tip 2: Split your input dataset into three independent subsets (training set, validation set, test set), and use the test set only once you complete training and optimization phases
- Tip 3: Frame your biological problem into the right algorithm category
- Tip 4: Which algorithm should you choose to start? The simplest one!
- Tip 5: Take care of the imbalanced data problem
- Tip 6: Optimize each hyper-parameter
- Tip 7: Minimize overfitting
- Tip 8: Evaluate your algorithm performance with the Matthews correlation coefficient (MCC) or the Precision-Recall curve
- Tip 9: Program your software with open source code and platforms
- Tip 10: Ask for feedback and help to computer science experts, or to collaborative Q&A online communities

Exercise and Code

- Linear Programming Methods

  - Link - https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l16-optimal/Optimization.ipynb

Course Project

Project Discussion: What Problem Fascinates You ?

- Data
  - Water
  - Finance
  - …
- Analytics
  - Search, Optimization, Learning, Planning, …
- Application
  - Building chatbot
- Users
  - Diverse demographics
  - Diverse abilities
  - Multiple human languages

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

Project Discussion: Dates and Deliverables

Project execution in sprints

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
  - Human interaction: Develop a basic chatbot (no AI), no problem focus
- Sprint 2: (Oct 10 – Nov 9)
  - Solving: Evaluate your solution on problem
  - Human interaction: Integrated your choice of chatbot (rule-based or learning-based) and methods
- Sprint 3: (Nov 14 – 30)
  - **Evaluation:** Comparison of your solver chatbot with an LLM-based alternative, like ChatGPT

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
- Nov 30, 2023
  - Project report due
- Dec 5 / 7, 2023
  - In-class presentation

Skeleton: A Basic Chatbot

- Run in an infinite loop until the user wants to quit
- Handle any user response
  - User can quit by typing “Quit” or “quit” or just “q”
  - User can enter any other text and the program has to handle it. The program should write back what the user entered and say – “I do not know this information”.
- Handle known user query types // Depends on your project
  - “Tell me about N-queens”, “What is N ?”
  - “Solve for N=4?”
  - “Why is this a solution? ”
- Handle chitchat // Support at least 5, extensible from a file
  - “Hi” => **“Hello”**
  - …
- ***Store session details in a file***

Illustrative Project

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Discussion: Illustration

1. Create a private Github repository called "CSCE58x-Fall2023-<studentname>-Repo". Share with Instructor (biplav-s) and TA (kausik-l)
2. Create Google folder called "CSCE58x-Fall2023-<studentname>-SharedInfo". Share with Instructor (prof.biplav@gmail.com) and TA (lakkarajukausik90@gmail.com)
3. Create a Google doc in your Google repo called "Project Plan" and have the following by next class (Sep 5, 2023)

1. **Title**: Solve and explain solving of n-queens puzzle
2. **Key idea**: Show students how a course project will look like
3. **Who will care when done**: students of the course, prospective AI students and teachers
4. **Data need**: n: the size of game; interaction
5. **Methods**: search
6. **Evaluation**: correctness of solution, quality of explanation, appropriateness of chat
7. **Users**: with and without AI background; with and without chess background
8. **Trust issue**: user may not believe in the solution, may find interaction offensive (why queens, not kings? …)

Project Illustration: N-Queens

- Sprint 1: (Sep 12 – Oct 5)
  - Solving: Choose a decision problem, identify data, work on solution methods
    - Method 1: Random solution
    - Method 2: Search – BFS
    - Method 3: Search - …
  - Human interaction: Develop a basic chatbot (no AI) as outlined
  - Deliverable
    - Code structure in Github
      - ./data
      - ./code
      - ./docs
      - ./test
    - Presentation: Make sprint presentation on Oct 12, 2023

Reference: Project Rubric

- **Project results** – 60%
  - Working system ? – 30%
  - Evaluation with results superior to baseline? – 20%
  - Considered related work? – 10%
- **Project efforts** – 40%
  - Project report – 20%
  - Project presentation (updates, final) – 20%

- **Bonus**
  - Challenge level of problem – 10%
  - Instructor discretion – 10%
- **Penalty**
  - Lack of timeliness as per announced policy (right) - up to 30%

Milestones and **Penalties**

- Oct 12, 2023
  - Project checkpoint
  - In-class presentation
  - **Penalty: presentation not ready by Oct 10, 2023 [-10%]**
- Nov 30, 2023
  - Project report due
  - **Project report not ready by date [-10%]**
- Dec 5 / 7, 2023
  - In-class presentation
  - **Project presentations not ready by Dec 4, 2023 [-10%]**

<Project Title> - <Your Name>

Format for Interim Presentation on Oct 12, 2023

Project Context

1. Problem
2. Who will care/ users
3. Data needs:
4. Methods:
5. Evaluation:
6. Trust issue:

Achievement

- Status
- Test Case
  - E.g., <input, correct output>
- Sample Result
- Discuss others points:
  - Challenges faced
  - Any help needed

1 min context, 1 min achievement, 1 min Q/A

Lecture 13: Summary

- We talked about
  - The variety of methods fpr classification
  - Logistic Regression
  - Decision trees
  - Random forest
  - Discussion: Choosing a method that works

Concluding Section

About Next Lecture – Lecture 14

Lecture 14: Machine Learning

- Structured Data: Supervised Methods
  - Naïve Bayes
  - Boosting
  - Explanation

- Reading material:
  - “Which ML to Use” with title: Data-driven advice for applying machine learning to bioinformatics problems
    https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5890912/
  - “10 tips with title”: Ten quick tips for machine learning in computational biology
    https://biodatamining.biomedcentral.com/articles/10.1186/s13040-017-0155-3