



Solving Problems with Decision Trees



TryEngineering

Provided by TryEngineering.org - www.tryengineering.org

Lesson Focus

This lesson focuses on how computing decision trees are used to specify and solve problems.

Lesson Synopsis

This lesson activity explores how simple computing concepts/algorithms have contributed to solving real life problems. Students will also learn solving problems with decision trees. Students will have the opportunity to work in teams to explore an example of how the decision tree can be used for detecting subscription fraud.

Age Levels

14-18

Objectives

Students will be able to:

- ◆ recognize a decision tree;
- ◆ recognize a problem where a decision tree can be useful in solving it;
- ◆ relate algorithms and decision trees, and be able to list some algorithms that can be matched to a decision tree"
- ◆ demonstrate teamwork and computing problem solving in groups.

Anticipated Learner Outcomes

As result of this activity, students should develop an understanding of:

- ◆ Decision trees
- ◆ Programming
- ◆ Algorithms

Lesson Activities

Students learn about decision trees, subscription fraud and how they can use decision trees to solve the subscription fraud problem. Students work in teams with specific task assigned to each member. The end result is a decision tree for detecting subscription fraud.

Resources/Materials

- ◆ Teacher Resource Documents (attached)
- ◆ Student Worksheets (attached)
- ◆ Student Resource Sheets (attached)

Internet Connections

- ◆ TryComputing (www.trycomputing.org).
 - ◆ ACM/IEEE Curricular task forces (<http://csta.acm.org/Curriculum/sub/ACMK12CSModel.html>).
 - ◆ Wikipedia: Decision trees (www.wikipedia.org)
 - ◆ Teamwork (Teamworkpm.net)
 - ◆ Wikipedia: Data analysis technique for fraud detection (www.wikipedia.org)
-

Recommended Reading

- ◆ Introduction to algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein ISBN: 978-0-262-03384-8.
 - ◆ Advances in security and payment methods for mobile commerce By Wen Chen Hu, Chung-Wei Lee, Weidong Kou ISBN: 1591403456
 - ◆ The decision tree by Kenneth A. Friedman ISBN: 0961586869
-

Optional Writing Activity

Write a decision tree on the type of decision software developers would need to consider when developing a game.

Solving Problems with Decision Trees



For Teachers: Teacher Resources

◆ Lesson Goal

The goal of this lesson is to explore the impact of computing in telecommunication. Students learn about using decision trees to detect subscription fraud on a single click, students will also learn how to work in teams.

◆ Lesson Objectives

- ◆ Learn about decision trees.
- ◆ Learn about algorithms: Decision tree is one way to display an algorithm.
- ◆ Learn about subscription fraud and how computing can help solve telecommunication challenges.
- ◆ Learn the implementation of algorithms using a programming language.
- ◆ Learn about teamwork and computing problem solving in groups.

◆ Materials

- ◆ Student resource sheet.
- ◆ Student worksheet.
- ◆ Internet access (If possible).

◆ Procedure

1. Show students the various Student Reference Sheets. These may be read in class, or provided as reading material.
2. Divide students into groups of 3 or 4. Teams can be named E.g. Fraud Detection Team 1 (or Team Microsoft programmers), Fraud Detection Team 2 (Team MAC Programmers)...Fraud Detection Team N (Team Google Programmers). The team naming will add more fun to the lesson.
3. Each team should assign tasks to its members. Members could also be named based on task assigned to the member. E.g.Member1: Your task is the analyze the manual procedure for detecting subscription fraud and build a decision tree from it hence he can be called the "Computer analyst/Team leader" of the team, Member2: Your task will be to build an algorithm in the form of if-else statements from the decision tree hence he can be called the "algorithm designer" of the team, Member3: Your task will be to illustrate how it will be implemented in a programming language" etc.
4. Teams should have the option of using Teamwork project management (www.teamworkpm.net) to complete the project. Note: Internet connectivity required. (Optional)
5. The "Team leader/Computer analyst" of the team should read about subscription fraud and write a short report briefing his other team members about the project.

Solving Problems with Decision Trees



For Teachers: Teacher Resources (continued)

6. All team members should work together to develop a step by step procedure on how they intend to achieve the goal. Starting from understanding how to detect subscription fraud manually, using this method to build a decision tree, translating the decision tree into if-else instructions, implementing it using PHP programming language and testing.
7. The First team member should also develop the decision tree for subscription fraud detection. Draw the decision tree, write a report on it, present & explain it to the other team members.
8. The Second team member should translate the decision tree into line by line conditional statements, write a report and explain it to he's other team members.
9. The third team member should investigate how this conditional statement can be translated into a program using any programming language and present he's report to he's team members.
10. Teams complete an evaluation/reflection worksheet, and present their findings to the class.

◆ Extended Option (Using teamworkpm.net)

If students decide to use teamworkpm.net

The team leader should open an account with teamworkpm.net using the team's name.

The team leader should invite he's other members using their email addresses to join the team online.

The team can now use teamworkpm.net to submit reports, communicate, and collaborate to build the program using the procedure above.

◆ Advanced Option

Have advanced students with knowledge in PHP to write a program using the decision tree to detect fraud.

◆ Time Needed

- ◆ Three to four 45 minute sessions.

Solving Problems with Decision Trees



Student Resource: Algorithm and Decision Trees

◆ Algorithms

In simple words an algorithm is a step-by-step procedure for calculations. In mathematics and computer science, an **algorithm** is an effective method expressed as a finite list of well-defined instructions for calculating a function. Algorithms are used for calculation, data processing, and automated reasoning. Starting from an initial state and initial input (perhaps empty), the instructions describe a computation that, when executed, will proceed through a finite number of well-defined successive states, eventually producing "output" and terminating at a final ending state. The transition from one state to the next is not necessarily deterministic; some algorithms, known as randomized algorithms, incorporate random input.

◆ Decision trees

Decision trees are used for deciding between several courses of action. Decision trees are predictive model, used to graphically organize information about possible options, consequences and end value. Decision trees are easy to understand and interpret. A decision tree is also a method for expressing algorithms. The goal of a decision tree is to create a model that predicts the value of a target variable based on several input variables.

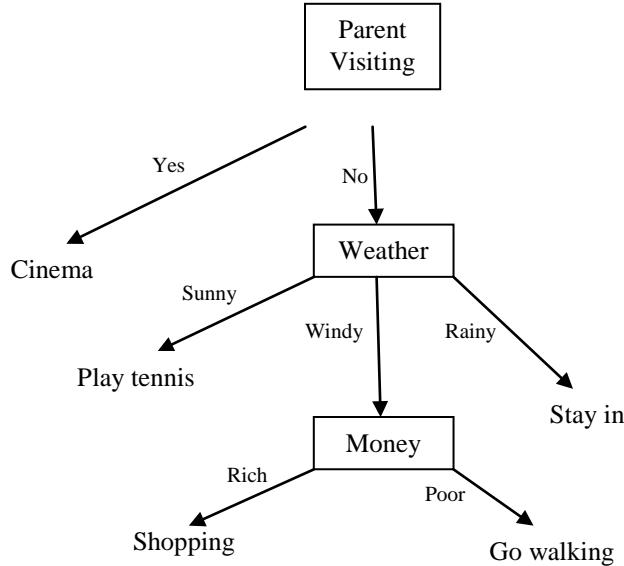
Imagine you only ever do four things at the weekend: go shopping, watch a movie, play tennis, go walking or just stay in. What you do depends on three things: the weather (windy, rainy or sunny); how much money you have (rich or poor) and whether your parents are visiting. You say to your yourself: if my parents are visiting, we'll go to the cinema. If they're not visiting and it's sunny, then I'll play tennis, but if it's windy, and I'm rich, then I'll go shopping. If they're not visiting, it's windy and I'm poor, then I will go walking. If they're not visiting and it's rainy, then I'll stay in.

To remember all this, you draw a flowchart which will enable you to read off your decision. We call such diagrams **decision trees**. A suitable decision tree for the weekend decision choices would be as follows:

Solving Problems with Decision Trees

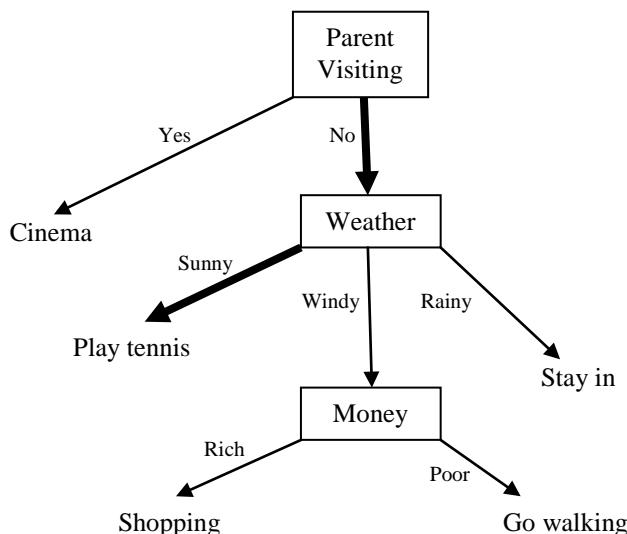


Student Resource: Algorithm and Decision Trees (continued)



We can see why such diagrams are called trees, because, while they are admittedly upside down, they start from a root and have branches leading to leaves (the tips of the graph at the bottom). Note that the leaves are always decisions, and a particular decision might be at the end of multiple branches (for example, we could choose to go to the cinema for two different reasons).

Armed with our decision tree, on Saturday morning, when we wake up, all we need to do is check (a) the weather (b) how much money we have and (c) whether our parent's car is parked in the drive. The decision tree will then enable us to make our decision. Suppose, for example, that the parents haven't turned up and the sun is shining. Then this path through our decision tree will tell us what to do:



Solving Problems with Decision Trees



Student Resource: Algorithm and Decision Trees (continued)

and hence we run off to play tennis because our decision tree told us to. Note that the decision tree covers all eventualities. That is, there are no values that the weather, the parents turning up or the money situation could take which aren't catered for in the decision tree. Note that, in this lecture, we will be looking at how to automatically generate decision trees from examples, not at how to turn thought processes into decision trees.

Reading Decision Trees

There is a link between decision tree representations and logical representations, which can be exploited to make it easier to understand (read) learned decision trees. If we think about it, every decision tree is actually a disjunction of implications (if ... then statements), and the implications are Horn clauses: a conjunction of literals implying a single literal. In the above tree, we can see this by reading from the **root node** (The topmost node of the tree) to each **leaf node** (the terminal nodes in the tree):

If the parents are visiting, then go to the cinema

or

If the parents are not visiting *and* it is sunny, then play tennis

or

If the parents are not visiting *and* it is windy *and* you're rich, then go shopping

or

If the parents are not visiting *and* it is windy *and* you're poor, then go walking

or

If the parents are not visiting *and* it is rainy, then stay in.

Of course, this is just a re-statement of the original mental decision making process we described. Remember, however, that we will be programming an agent to learn decision trees from example, so this kind of situation will not occur as we will start with only example situations. It will therefore be important for us to be able to read the decision tree the agent suggests.

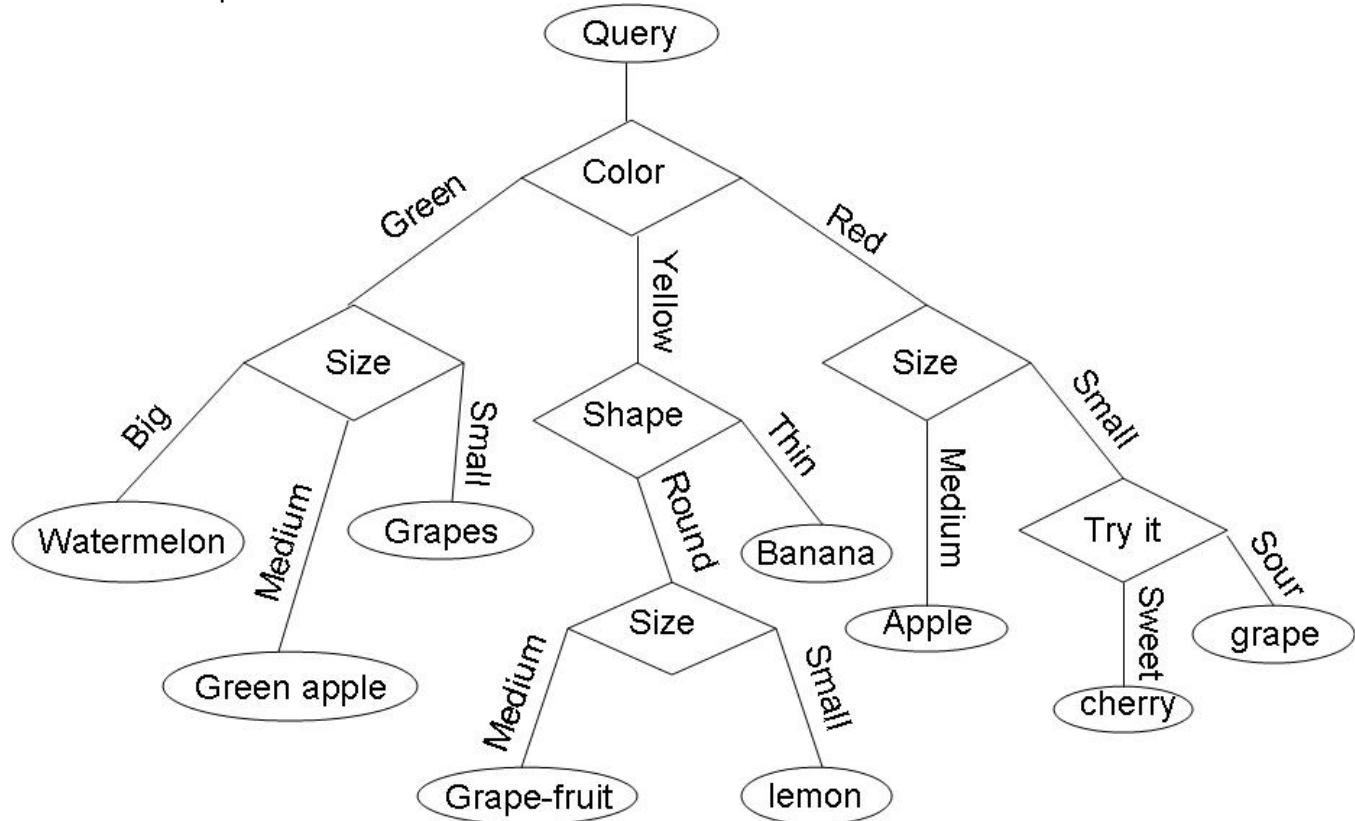
Decision trees don't have to be representations of decision making processes, and they can equally apply to categorization problems. If we phrase the above question slightly differently, we can see this: instead of saying that we wish to represent a decision process for what to do on a weekend, we could ask what kind of weekend this is: is it a weekend where we play tennis, or one where we go shopping, or one where we see a film, or one where we stay in, or go walking?

Solving Problems with Decision Trees



Student Resource: Algorithm and Decision Trees (continued)

Another example:



Solving Problems with Decision Trees



Student Resource: What is Subscription Fraud?

◆ Telecommunication fraud



Telecommunication fraud is the theft of telecommunication service (telephones, cell phones, computers etc.) or the use of telecommunication service to commit other forms of fraud. Victims include consumers, businesses and communication service providers. Subscription fraud is a type of telecommunication fraud

◆ Subscription fraud

The main form of telecommunication fraud that has taken place to date is subscription fraud (that is, the failure to pay for services used). Detection of such fraud is difficult because it is easily masked as bad debt. In subscription fraud, offenders typically misrepresent their identity in order to avoid payment. Misrepresentation of identity is also important because in the most severe cases, subscription fraud is not an end in itself but rather a platform for several other frauds.

Subscription fraud is the act of using telephone services without the intention of paying. Customers subscribe for postpaid services, when the time comes for customers to pay they don't. The telecommunication industry has lost billions of dollars to subscription fraud.

The table below shows an example of how a customer billing database looks like;

Name	Phone no.	Customer Segment	Outstanding Balance	Unbilled	Credit Limit	Last Date of payment
John	12345678	Consumer	\$836	\$1003	\$667	31/8/2011
Gate Corporation	12435566	Corporate	\$3082	\$2027	\$6667	8/8/2011
Michael	23767868	Consumer	\$70	\$300	\$500	3/7/2011
Binary Enterprise	34556578	Enterprise	\$10000775	\$708000	\$6000000	16/8/2011

Solving Problems with Decision Trees



Student Resource: What is Subscription Fraud? (continued)

Name: Names of the customer

Phone number: Indicates the phone number of the customer

Customer Segment: Category of customer, either corporate, consumer or enterprise. Corporate is for an organization, consumer is directly for the customer while enterprise is for larger organizations.

Outstanding Balance: Used credits

Unbilled: Unused credit

Credit Limit: Credit limit allocated. Credit limit is also the maximum amount of credit that is supposed to be used by the customer.

Last Date of payment: As the name implies, it is the last date the customer made payments.

Note: These are fictional data. This data does not exist. It is being used for illustration purposes. It is similar to how a telecommunication billing database for postpaid customers looks like. Although it varies among several telecom companies, but these are the basic attributes that can be seen in the database.

Solving Problems with Decision Trees



Student Resource: Detecting Subscription Fraud?

From investigations the telecommunication industry discovered a way for detecting subscription fraud. The procedure is illustrated below; Two new variables were created; Pay Duration (in days) and ratio.

Pay Duration: Date of checking customer's fraud status (today's date)-last date of payment by customer

Ratio: balance/credit limit. Where balance=Outstanding balance + unbilled



Investigations showed that whenever the pay duration is greater than 31 days the customer is likely to be involved in BAD DEBT.

But if it's less than or equal to 31 days, the ratio of the customer is checked, if the ratio is less than 1 then the customer's fraud status is NOT FRAUD, but if the ratio is greater than 1, the pay duration is checked again, if the pay duration is less than or equal to 0days then the customers fraud status is SUBSCRIPTION FRAUD, but if its greater than 0days it is BAD DEBT.

A close study of the manual procedure for detecting subscription should make sense.

Using this procedure we can reproduce the table above and fill in the last column (Fraud status) as of 31st August 2011 i.e. Assume we were checking the fraud status on this date.

Name	Phone no.	Customer Segment	Outstanding Balance	Unbilled	Credit Limit	Last Date of payment	Fraud Status
John	12345678	Consumer	\$836	\$1003	\$667	31/8/2011	Subscription fraud
Gate Corporation	12435566	Corporate	\$3082	\$2027	\$6667	8/8/2011	Not fraud
Michael	23767868	Consumer	\$70	\$300	\$500	3/7/2011	Bad Debt
Binary Enterprise	34556578	Enterprise	\$10000775	\$708000	\$6000000	16/8/2011	Bad debt

Solving Problems with Decision Trees



Student Resource: Detecting Subscription Fraud?

◆ Teamwork/Project Management

Teamwork:

Project management:

This lesson has introduced group working. Software systems are not built by individuals, but by large teams. More advanced groups may be interested in considering how they could manage decision making within the context of building large software systems and thinking about the many decision this would involve. Teams don't need to be meeting all team members very frequently because most of the work can be done online. Because of the importance of teamwork, computer scientists are developing different simulation programs for teamwork to be achieved online.

Solutions

Teamworkpm.net

Teamworkpm.net is one of the numerous online project management software available online for computer scientist, managers etc to use to achieve their goals. Some companies already using the teamworkpm.net includes;

Features:

- Task management.
- Teamwork.
- Quick overview of recent activity.
- Easy access to all your projects.
- Keep team informed with announcements.
- Online communication among team members.
- Assign tasks to team members

Below are some clips taking from teamworkpm.net.

The screenshot shows the 'Digital Crew Project Manager' dashboard. At the top, there's a navigation bar with links to Dashboard, Everything, Projects, Calendar, and Statuses. Below the header, the title 'Digital Crew Project Manager' is displayed above a logo for 'DIGITAL CREW'. On the left, there's a sidebar titled 'Your projects' which lists several projects under the heading 'Digital Crew': Bru Hostel, Cescie website, Community AkariSoftware.co, Connolly Insulation, Cork Business Association, Cork CEE Website, CorkLetting.com, Cost Reduction Alliance, and County Cork VEC website. The main content area is titled 'Latest activity' and shows a feed of comments and messages. One comment from 'Hi Dan' is visible, followed by a message from 'CBA Website' and another comment from 'Hi Dan'. The interface is clean and modern, designed for managing multiple projects simultaneously.

Solving Problems with Decision Trees



Student Resource: Detecting Subscription Fraud? (continued)

The screenshot shows the Teamwork Project Manager interface. At the top, there's a navigation bar with links for Overview, Tasks, Milestones, Messages, Files, Time, Notebook, Risk Register, and Resources. The main area is titled 'Project Overview' and shows an 'Upcoming Milestones' calendar. The calendar grid shows tasks for weeks 40, 41, and 42. Week 40 includes tasks like 'Roll out latest update' and 'Comparing call & texts with marketing results in quickbird'. Week 41 includes tasks like 'notify SMS access to everyone who is selected to be notified of an event' and 'File: display order all over shop after File upload'. Week 42 includes tasks like 'File: display all over the shop' and 'File: display all over the shop'. Below the calendar, there are sections for 'Today' and 'Yesterday' with their respective task lists.

The screenshot shows the 'Project Settings' page in the Teamwork Project Manager. The page has a header with links for Dashboard, Everything, Projects, Calendar, and Statuses, and a 'Switch Project' dropdown. The main content area is titled 'Teamwork Project Manager Project Settings'. It contains three main sections: 'Project Settings' (with fields for Project Name, Description, and Contact Details), 'Project Announcement' (with an announcement message and an 'Enabled' checkbox), and 'Project Tabs' (with checkboxes for Tasks, Messages, Time, Risk Register, and Billing).

Instructions for using teamworkpm.net:

Teamworkpm.net is very easy to use. Visit the website and follow the instructions.

Mercurial (mercurial.selenic.com):

Another kind of decision that teams need to make is about how they organize individuals in the team to collaboratively work on the same documents without loosing one another's work by editing the same documents at the same time. More advanced computer scientist use software like the mercurial source control management. Mercurial is a free, distributed source control management tool. It offers you the power to efficiently handle projects of any size while using an intuitive interface. It is easy to use and hard to break, making it ideal for anyone working with versioned files. Visit website to learn more. Such systems effectively encode a decision making protocol in their source code to ensure that teams can work collaboratively without risk of accidentally overwriting the work of other team members.

Solving Problems with Decision Trees



Student Worksheet:

Design Your Subscription Fraud Detection Program

You are a team of computer scientist and have been approached by a telecommunication company to develop a program where they can check the fraud status. The Information Technology Department of the company requires you to send them the decision tree and algorithm (Conditional Statements) you intend to use.

You are to work as a team, and have the option of using teamworkpm.net.

Study your student resource and use it to write the algorithm and draw your decision tree.

Assign tasks to members in your team and use the report sheet for feedback.

Below is an example of the report sheet which each member of the team should have.

Report Sheet

Team:

Member:

Date & Time:

Report:

Solving Problems with Decision Trees



Student Worksheet: Design Your Subscription Fraud Detection Program

◆ Evaluation Phase

1. Use your program to check the status of these customers Note: Students should assume they are conducting the analysis on 1/1/2012. Fill in the last column

Name	Phone no.	Customer Segment	Outstanding Balance	Unbilled	Credit Limit	Last Date of payment	Fraud Status
Mary	12345678	Consumer	\$467	\$1003	\$6633	23/1/2012	
Kings Palace Hotel	12435566	Corporate	\$30	\$203273	\$66673	16/12/2011	
Prof. John Gandhi	23767868	Consumer	\$703	\$300	\$500	1/1/2012	
Senator Smith	34556578	Consumer	\$1000	\$70800	\$7500	31/12/2012	

2. What is the difference between subscription fraud and bad debt?

3. Draw a simple decision tree for your tomorrow's schedule.

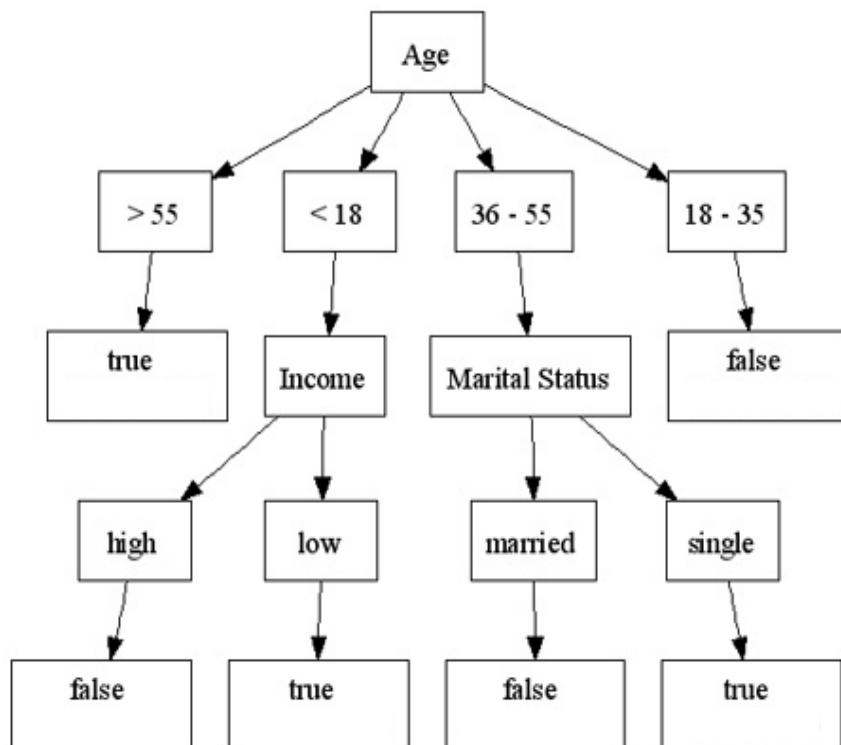
Solving Problems with Decision Trees



Student Worksheet: Design Your Subscription Fraud Detection Program

4. Draw a simple decision tree to categorize animals (mammal, fish, reptile, bird etc) using physical attributes (whether it lays eggs, number of legs, etc.). This could easily be phrased as a question of learning a decision tree to decide which category a given animal is in, e.g., if it lays eggs and is homeothermic, then it's a bird, and so on...

5. Translate the decision tree below into if/else (Conditional) statements;



Solving Problems with Decision Trees



Teacher Resource:

Alignment to Curriculum Frameworks

Note: Lesson plans in this series are aligned to one or more of the following sets of standards:

- U.S. Science Education Standards (http://www.nap.edu/catalog.php?record_id=4962)
- U.S. Next Generation Science Standards (<http://www.nextgenscience.org/>)
- International Technology Education Association's Standards for Technological Literacy (<http://www.iteea.org/TAA/PDFs/xstdnd.pdf>)
- U.S. National Council of Teachers of Mathematics' Principles and Standards for School Mathematics (<http://www.nctm.org/standards/content.aspx?id=16909>)
- U.S. Common Core State Standards for Mathematics (<http://www.corestandards.org/Math>)
- Computer Science Teachers Association K-12 Computer Science Standards (<http://csta.acm.org/Curriculum/sub/K12Standards.html>)

◆ Principles and Standards for School Mathematics

Data Analysis and Probability Standard

- ◆ Select and use appropriate statistical methods to analyze data

Problem Solving Standard

As a result of activities, all students should develop

- ◆ Apply and adapt a variety of appropriate strategies to solve problems.
- ◆ Monitor and reflect on the process of mathematical problem solving.

Representation Standard

- ◆ Create and use representations to organize, record, and communicate mathematical ideas
- ◆ Use representations to model and interpret physical, social and mathematical phenomena

◆ Common Core State Standards for Mathematics Grades 3-8 (Ages 8-14)

Statistics & Probability

- Investigate chance processes and develop, use, and evaluate probability models.
 - ◆ CCSS.Math.Content.7.SP.C.8 Find probabilities of compound events using organized lists, tables, tree diagrams, and simulation.

◆ Common Core State Standards for Mathematics Grades 8-14 (Ages 14-18)

Statistics and Probability

- Using Probability to make decisions
 - o Use probability to evaluate outcomes of decisions
 - ◆ CCSS.Math.Content.HSS-MD.B.6 (+) Use probabilities to make fair decisions (e.g., drawing by lots, using a random number generator).

◆ Standards for Technological Literacy – All Ages

The Nature of Technology

- ◆ Standard 1: Students will develop an understanding of the characteristics and scope of technology.
- ◆ Standard 2: Students will develop an understanding of the core concepts of technology.
- ◆ Standard 3: Students will develop an understanding of the relationships among technologies and the connections between technology and other fields of study.

Solving Problems with Decision Trees



Teacher Resource:
Alignment to Curriculum Frameworks

◆ Standards for Technological Literacy – All Ages

Technology and Society

- ◆ Standard 4: Students will develop an understanding of the cultural, social, economic, and political effects of technology.

The Designed World

- ◆ Standard 17: Students will develop an understanding of and be able to select and use information and communication technologies.

◆CSTA K-12 Computer Science Standards Grades 6-9 (ages 11-14)

5. 2 Level 2: Computer Science and Community (L2)

- ◆ Computational Thinking (CT)
 - 1. Use the basic steps in algorithmic problem-solving to design solutions
 - 3. Define an algorithm as a sequence of instructions that can be processed by a computer.
 - 8. Use visual representations of problem states, structures, and data (e.g., graphs, charts, network diagrams, flowcharts).
 - 9. Interact with content-specific models and simulations (e.g., ecosystems, epidemics, molecular dynamics) to support learning and research.
- ◆ Collaboration (CL)
 - 1. Apply productivity/multimedia tools and peripherals to group collaboration and support learning throughout the curriculum.
 - 3. Collaborate with peers, experts, and others using collaborative practices such as pair programming, working in project teams, and participating in group active learning activities.
 - 4. Exhibit dispositions necessary for collaboration: providing useful feedback, integrating feedback, understanding and accepting multiple perspectives, socialization.
- ◆ Computing Practice & Programming (CPP)
 - 2. Use a variety of multimedia tools and peripherals to support personal productivity and learning throughout the curriculum.
 - 4. Demonstrate an understanding of algorithms and their practical application.
 - 7. Identify interdisciplinary careers that are enhanced by computer science.

◆CSTA K-12 Computer Science Standards Grades 9-12 (ages 14-18)

5.3 Level 3: Applying Concepts and Creating Real-World Solutions (L3)

5.3.A Computer Science in the Modern World (MW)

- ◆ Community, Global, and Ethical Impacts (CI)
 - 2. Discuss the impact of computing technology on business and commerce (e.g., automated tracking of goods, automated financial transactions, e-commerce, cloud computing).

Solving Problems with Decision Trees



Teacher Resource:
Alignment to Curriculum Frameworks

◆CSTA K-12 Computer Science Standards Grades 9-12 (ages 14-18)

5.3 Level 3: Applying Concepts and Creating Real-World Solutions (L3)

5.3.B Computer Science Concepts and Practices (CP)

- ◆ Computational Thinking (CT)
 - 9. Analyze data and identify patterns through modeling and simulation.
- ◆ Collaboration (CL)
 - 1. Use project collaboration tools, version control systems, and Integrated Development Environments (IDEs) while working on a collaborative software project.