Unit II

Machine Learning

Introduction
What Is Machine Learning?

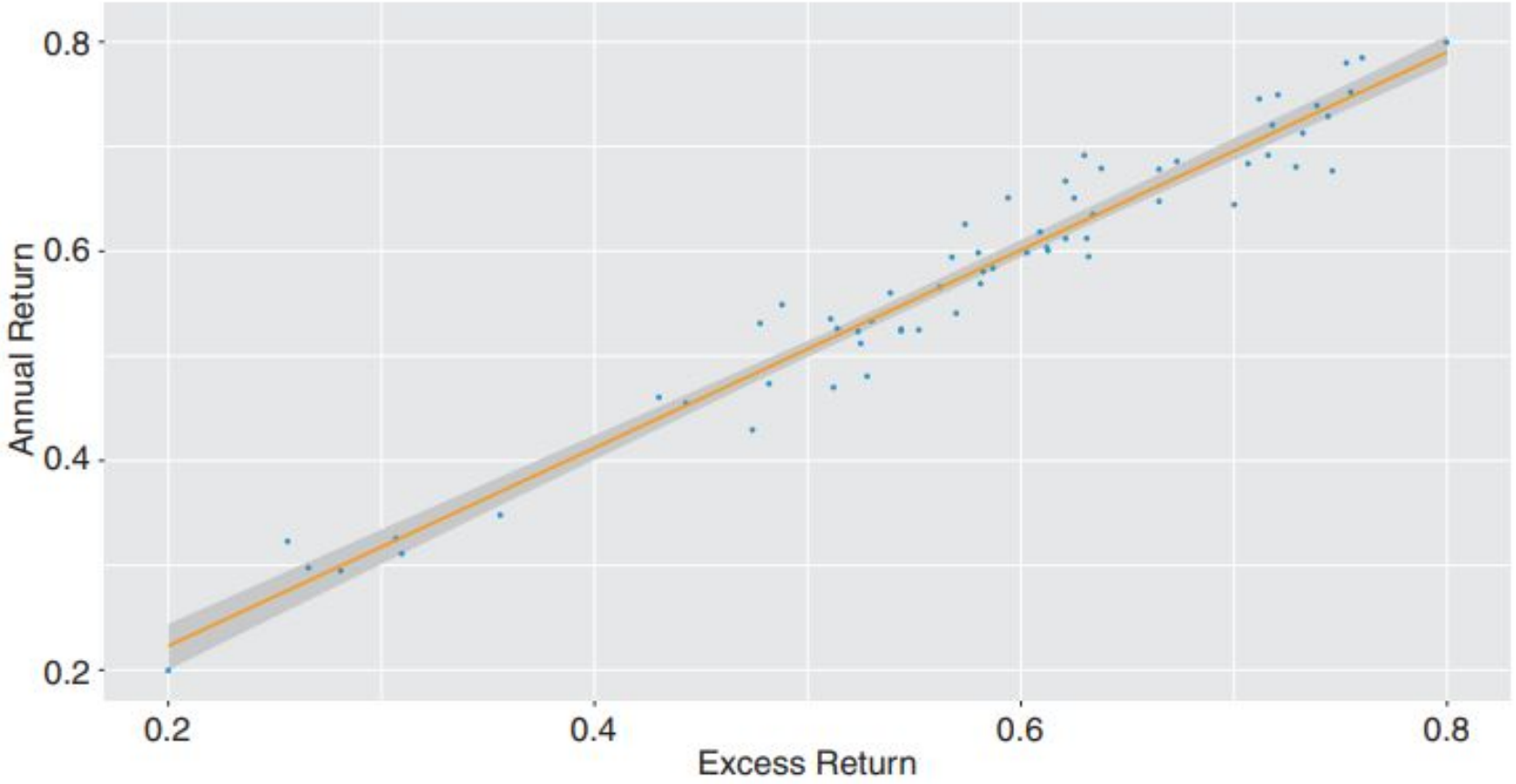
- A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.
- to create a good machine learning system
  - Data preparation capabilities.
  - Algorithms– basic and advanced.
  - Automation and iterative processes.
  - Scalability.
  - Ensemble modeling.
- Types of machine learning algorithms
  - Supervised learning. When we know the labels on the training examples we are using to learn.
  - Unsupervised learning. When we do not know the labels (or even the number of labels or classes) from the training examples we are using for learning.
  - Reinforcement learning. When we want to provide feedback to the system based on how it performs with training examples.

Regression

- Eg : predict someone's income based on their education. learning such a relationship is regression
- modeling the relationship between variables of interest
- see how a variable X affects a variable y. Here, X is called the independent variable or predictor; y is called the dependent variable or response.
- Some of the most popular regression algorithms are:
  - Ordinary least squares regression (OLSR)
  - Linear regression
  - Logistic regression
  - Stepwise regression
  - Multivariate adaptive regression splines (MARS)
  - Locally estimated scatterplot smoothing (LOESS)

Regression

- to predict the annual return using excess return of stock in a stock portfolio



An example showing a relationship between annual return and excess return of stock using linear regression from the stock portfolio dataset.[9]

Regression

- Draw a random line and see how well it fits the data.
- For this, we can find the distance of each data point from that line and add it all up.
- That gives a number, often called cost or error.
- Now, let us draw another line and repeat the process.
- We get another number for cost.
- If this is lower than the previous one, the new line is better.
- Keep repeating this until we find a line that gives us the lowest cost or error,
- Once found our most fitting line □ solving the problem of regression.
- Gradient descent help to find such a model.

Multilinear Regression

- allows for the modeling of relationships between a dependent variable and multiple independent variables.
- widely used statistical technique in data analysis, economics, finance, and various scientific discipline
- In multilinear regression, the relationship between the dependent variable Y and the independent variables $X1, X2, \ldots, Xn$ is modeled as a linear combination of these variables:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

- Y is the dependent variable.
- $X_1, X_2, \ldots, X_n$ are the independent variables.
- β_0 is the y-intercept (the constant term).
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients that represent the impact of each independent variable on the dependent variable.

Multilinear Regression - example

- Consider a real-world example where a company wants to predict its monthly sales based on several factors, such as advertising expenditure, price discounts, and the number of sales representatives. Here, the dependent variable Y is the monthly sales, and the independent variables $X1$,$X2$,$X3$ represent advertising expenditure, price discounts, and the number of sales representatives, respectively.

- The multilinear regression model for this scenario could be written as:

- Sales=β0+β1(Advertising Expenditure)+β2(Price Discounts)+β3(Number of Sales Representatives)

Gradient Descent

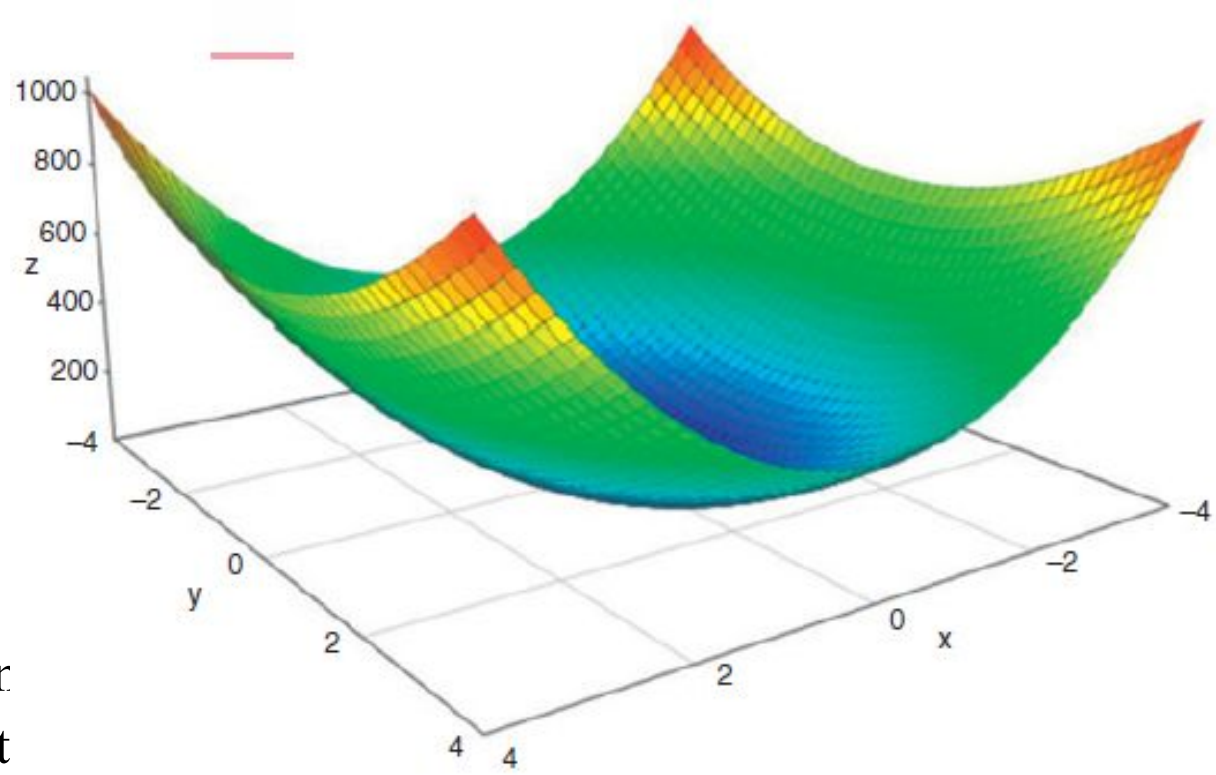
- to fit multiple lines to the same dataset, each represented by the same equation but with different m and b values
- The idea is to find the best one to represent the dataset better than the other lines. That is, we need to find the best set of m and b values
- To define an error function (cost function) to measure how good a given line is. This function will take in a (m, b) pair and return an error value based on how well the line fits our data.
- Error function is given as follows :

$$\epsilon = \frac{1}{n}\sum_{i=1}^{n}((mx_i + b) - y_i)^2.$$

- squared the distance to ensure that it is positive and to make the error function differentiable

Gradient Descent

- The error function is defined to lower the error values. Minimal values best line.
- Since error function consists of two parameters (m and b), we can visualize it as a 3D
- surface.



- x represents slope, y represents intercept ar
- The height of the function at each point is t
- Blue color indicates the lower the error function value and the better it fits our data.
- So, to find the best m and b set that will minimize the cost function gradient descent

Gradient Descent

- Gradient descent is an approach to find minima – points where the error is at its lowest.
- When we run a gradient descent search, we start from some location on this surface and move downhill to find the line with the lowest error.
- To run gradient descent on this error function, we first need to compute its gradient or slope.
- To compute it, we will need to differentiate the error function.
- Since the function is defined by two parameters (m and b), we need to compute a partial derivative for each.

$$\frac{\partial \epsilon}{\partial m} = \frac{2}{n}\sum_{i=1}^{n}((mx_i + b) - y_i)\frac{\partial}{\partial m}((mx_i + b) - y_i)$$
$$= \frac{2}{n}\sum_{i=1}^{n}((mx_i + b) - y_i)x_i$$

$$\frac{\partial \epsilon}{\partial b} = \frac{2}{n}\sum_{i=1}^{n}((mx_i + b) - y_i)\frac{\partial}{\partial b}((mx_i + b) - y_i)$$
$$= \frac{2}{n}\sum_{i=1}^{n}((mx_i + b) - y_i).$$

Gradient Descent

- Now, start at any pair of m and b values (i.e., any line) and let the gradient descent algorithm march downhill on the error function toward the best line.
- Each iteration will update m and b to a line that yields a slightly lower error than the previous iteration.
- The direction to move in for each iteration is calculated using the two partial derivatives from the above two equations

- To update m and b,
  - $m = m - \alpha \frac{\delta \mathcal{E}}{\delta m}$
  - $b = b - \alpha \frac{\delta \mathcal{E}}{\delta b}$

- Where α is the learning rate, a small positive number that controls how large a step you take in each iteration

Gradient Descent

- Generalize :
- In the above case, we have two parameters m and b.
- There could be many parameters in a problem, depending on the dimensionality of the data or the number of features available □ θ □ to estimate the best possible values of the θ.
- a model that could have any number of parameters and represent it using h(hypothesis function)
- h(x) is the predicted output.
- θi represents the parameters (weights) of the model.
- xi represents the features of the input data.
  (1)
- n is the number of features
- consider θ0 = b, θ1 = m, and assign x0 = 1, (for 2 parameters)

$$h(x) = \sum_{i=0}^{n} \theta_i x_i.$$

Gradient Descent

- Define a cost (error) function for the previous hypothesis □ to measure how well the hypothesis function fits the training data. It represents the average squared difference between the predicted values and the actual target values over all training examples.

- $$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h(x^i) - y^i)^2. \qquad (2)$$

$J(\theta)$ is the cost function.

m is the number of training examples.

$h(x^{(i)})$ is the predicted value for the i-th training example.

$y^{(i)}$ is the actual output (target value) for the i-th training example.

- The cost function $J(\theta)$ in Equation (2) is derived by calculating the error between the predicted output $h(x(i))$ and the actual output $y(i)$ for all training examples.

Gradient Descent

- Derivation of Equation (2)

1. **Substitute the Hypothesis Function $h(x)$:**
   - The hypothesis function $h(x)$ in Equation (1) is used to predict the output for each training example.
   - For the i-th training example, the prediction is $h(x^{(i)}) = \sum_{j=0}^{n} \theta_j x_j^{(i)}$.
2. **Calculate the Error:**
   - The difference (error) between the predicted value $h(x^{(i)})$ and the actual value $y^{(i)}$ is $\left(h(x^{(i)}) - y^{(i)}\right)$.
3. **Square the Error:**
   - The squared error for each training example is $\left(h(x^{(i)}) - y^{(i)}\right)^2$.
4. **Sum Over All Training Examples:**
   - Sum the squared errors over all m training examples: $\sum_{i=1}^{m} \left(h(x^{(i)}) - y^{(i)}\right)^2$.
5. **Compute the Average and Apply the Factor $\frac{1}{2m}$:**
   - The cost function $J(\theta)$ is the average of the squared errors across all training examples, with an additional factor of $\frac{1}{2m}$ for convenience in gradient descent calculations. This gives the final cost function in Equation (2).

Gradient Descent

- Minimize $J(\theta)$ □ for all parameters.
- Let us take for one parameter θj
- Step 1 : compute the partial derivative of J (θ) with respect to θj:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{\partial}{\partial \theta_j}\left(\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2\right)$$

- Step 2 : Differentiate Using the Chain Rule
- Differentiate the sum inside the cost function. The cost function is a sum of squared errors, and each error term depends on θi through the hypothesis function
- $$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)\frac{\partial h_\theta(x^{(i)})}{\partial \theta_j} \qquad \text{--(3)}$$

The term $\left(h_\theta(x^{(i)}) - y^{(i)}\right)$ is the error for the i-th example.

We multiply the error by the derivative of the hypothesis $h_\theta(x^{(i)})$ with respect to θ_j because we're computing the gradient with respect to θ_j.

Gradient Descent

- Step 4 : Compute the Derivative of the Hypothesis

$$h_\theta(x^{(i)}) = \sum_{k=0}^{n} \theta_k x_k^{(i)}$$

- Where *n* is the number of features, and $x_k^{(i)}$ is the *k*th feature of the *i*th example.

The derivative of $h_\theta(x^{(i)})$ with respect to θ_j is simply:

$$\frac{\partial h_\theta(x^{(i)})}{\partial \theta_j} = x_j^{(i)}$$

- Substitute this in (3) (gradient)

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

Gradient Descent

- Step 5 : Update the Parameters Using Gradient Descent
- Gradient descent algorithm updates each parameter θj by moving it in the direction of the negative gradient:

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

- Where:α is the learning rate, a small positive number that controls how large a step you take in each iteration.
- The gradient $\frac{\partial J(\theta)}{\partial \theta_j}$ tells you the direction in which to adjust θj to decrease the cost function.

Gradient Descent

- Substituting the gradient derived:

$$\theta_j := \theta_j - \alpha \left(\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x_j^{(i)} \right)$$

- The above equation updates each parameter θj by moving it in the direction that decreases the cost function $J(\theta)$.

kNN (k Nearest Neighbor)

- A real life example :
- A healthcare provider wants to predict whether a patient has a particular disease (e.g., diabetes) based on their health parameters such as glucose level, blood pressure, and BMI.
- The hospital has a dataset of previous patients' health records, along with whether they were diagnosed with diabetes.
- Dataset: The dataset contains features like glucose level, blood pressure, BMI, and the diagnosis (1 = diabetes, 0 = no diabetes).
- Each record represents a patient.

kNN (k Nearest Neighbor)

- Dataset

The dataset contains features like glucose level, blood pressure, BMI, and the diagnosis (1 = diabetes, 0 = no diabetes). Each record represents a patient.

Glucose Level	Blood Pressure	BMI	Diagnosis (Output)
85	75	22.0	0
90	80	23.5	0
140	90	28.5	1
160	95	30.0	1
110	85	26.0	0
155	100	29.0	1

- **Task:**
- A new patient comes in with the following health parameters:
- Glucose Level: 150, Blood Pressure: 92, BMI: 28.0
- Predict whether this new patient has diabetes.

kNN (k Nearest Neighbor)

- A classification algorithm
- Classification can be supervised or unsupervised.
- When there are only two choices, it is called two-class or binomial classification
- When there are more categories, it is known as multiclass or multinomial classification.
- Steps in kNN.
  1. As in the general classification problem, we have a set of data points for which we know the correct class labels.
  2. When we get a new data point, we compare it to each of our existing data points and find similarity.
  3. Take the most similar k data points (k nearest neighbors).
  4. From these k data points, take the majority vote on their labels. The winning label is the label/class of the new data point.
- The number k is usually small between 2 and 20.
- The more the number of nearest neighbors (value of k), the longer it takes us to do the processing.

Decision Tree

- used for classification problems.
- goal is to create a model that predicts the value of a target variable based on several input variables.
- A decision tree builds classification or regression models in the form of a tree structure.
- It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
- The final result is a tree with decision nodes and leaf nodes.
- Several algorithms exist that generate decision trees, such as ID3/4/5, CART, CLS, etc.
- ID3 employs Entropy and Information Gain to construct a decision tree.
- **Entropy:** Entropy (E) measures disorder, uncertainty, or randomness. If a coin is tossed , there is an equal chance of getting a head and a tail. In other words, we would be most uncertain about the outcome, or we would have a

$$\text{Entropy}(E) = -\sum_{i=1}^{k} p_i \log_2(p_i).$$

- K is the number of possible class values, and pi is the number of occurrences of the class i=1 in the dataset.

Decision Tree

Information Gain

- Information Gain in decision trees is used to quantify how much a particular feature helps to reduce the uncertainty or entropy about the target class after splitting the dataset based on that feature.
- It measures the effectiveness of an attribute in classifying the data.
- Information Gain is the difference between the entropy of the original dataset and the weighted entropy after splitting the dataset t based on a feature.

$$\text{Information Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \times \text{Entropy}(S_v)$$

- Where:*S* is the dataset.
- A is the feature used to split the dataset.
- *Sv* is the subset of the dataset where the feature *A* takes the value *v*
- **Entropy(S) is** the entropy of the dataset before the split. Entropy measures the impurity in a dataset. The higher the entropy, the less pure the dataset.
- **Information Gain** tells us how much a feature reduces the uncertainty in predicting the target class. A higher information gain indicates that the feature better separates the classes and is a good candidate for splitting the node in the decision tree.
- Example for Decision Tree : example

Naïve Bayes

- Bayesian classification represents a supervised learning method and statistical method for classification.
- It is a classification technique based on Bayes' theorem with an assumption of independence among predictors.
- Here, all attributes contribute equally and independently to the decision.
- Naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature
- Bayes' theorem, which provides a way of calculating posterior probability P(c| x) from P(c), P(x), and P(x| c).

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}.$$

- P(c| x) is the posterior probability of class (c, target) given predictor (x, attributes).
- P(c) is the prior probability of class.
- P(x| c) is the likelihood, which is the probability of the predictor given class
- P (x) is the prior probability of the predictor

$$P(c|x) = \frac{P(x_1|c)P(x_2|c)P(x_3|c)P(x_4|c)\ldots P(x_n|c)P(c)}{P(x)}$$

where $x_1, x_2, x_3, \ldots, x_n$ are independent priors.

Workout example – Naïve Bayes

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

Example

Clustering

- Unsupervised learning, data points have no labels associated with them.
- The goal of an unsupervised learning algorithm is to organize the data in some way or to describe its structure.
- Clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense
- Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields.
- types of clustering algorithms :
  - agglomerative (going bottom to top) building clusters or groups of similar data points from individual data points.
  - divisive (going top to bottom) works in a top-down mode, where the goal is to break up the cluster containing all objects into smaller clusters

Divisive Clustering : kMeans clustering

1. determine the number of clusters (k) and assume the centroid or center of these clusters.

Take any random objects as the initial centroids, or the first k objects in sequence can also serve as the initial centroids.

2. Then the k-means algorithm will do the three steps below until convergence.

i. Begin with a decision on the value of k = number of clusters.

ii. Put any initial partition that classifies the data into k clusters. Assign the training samples randomly or systematically, as in the following:

a. Take the first k training sample as single-element clusters.

b. Assign each of the remaining (N − K) training samples to the cluster with the nearest centroid.

c. After each assignment, recompute the centroid of the gaining cluster.

iii. Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

Example - kMeans Clustering

Individual	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Workout example

- Step 1 : The individuals 1 and 4 are the furthest apart, making them ideal candidates for partitioning. (centers of two different clusters, or centroids)

	Individual	Mean vector (centroid)
Cluster 1	1	(1.0, 1.0)
Cluster 2	4	(5.0, 7.0)

- Step 2 :

Kmeans clustering

Expectation-Maximization

- **Expectation-Maximization** is a powerful statistical technique used to find maximum likelihood estimates of parameters in models that involve latent (hidden) variables.
- It's especially useful in scenarios where data is incomplete or has missing values, and direct computation of parameters is difficult.
- **How Does It Work?**
- Think of EM like this:
- You have a puzzle, but some pieces are hidden. You want to complete the puzzle.
- You start with a guess for where the hidden pieces might be.
- Then, you follow a two-step process over and over to improve your guess until the puzzle looks right.

Expectation-Maximization

- . **1. E-Step (Expectation Step):**
- In this step, **estimate** what the hidden pieces (or missing data) might be based on current understanding. It’s like saying, "Okay, I don’t know exactly what the missing parts are, but based on what I know so far, I’ll guess they probably look like this."
- **2. M-Step (Maximization Step):**
- In this step, after making guesses, **improve** the model by adjusting the known pieces to better fit with the guessed hidden ones. This is like fine-tuning puzzle pieces to fit the guessed shapes better.

Expectation-Maximization

- **Simple Example: Finding a Hidden Coin Bias**
- Imagine flipping a coin, but don't know if it's a fair coin (50/50 heads or tails). The coin is flipped several times, but sometimes, the result is hidden. We can see some flips but not all.
- How to use EM:
- **Step 1 (Guess)**: Start by guessing the probability that the coin gives heads (say, we guess it's a biased coin with 60% heads).
- **Step 2 (Improve)**: Based on the flips we can see, figure out how likely our guess is (e.g., if we see more tails than expected, we might adjust your guess for heads down a little).
- **Repeat**: we keep guessing and improving until our estimate for the heads/tails ratio stops changing much.

Expectation-Maximization

- **Why Use EM?**
- EM is used when we don't have complete information but still want to make the best possible guess. It's commonly used in situations like:
- **Clustering Data**: Grouping similar items when we don’t know which group they belong to.
- **Medical Imaging**: Identifying patterns in scans where some information is unclear.

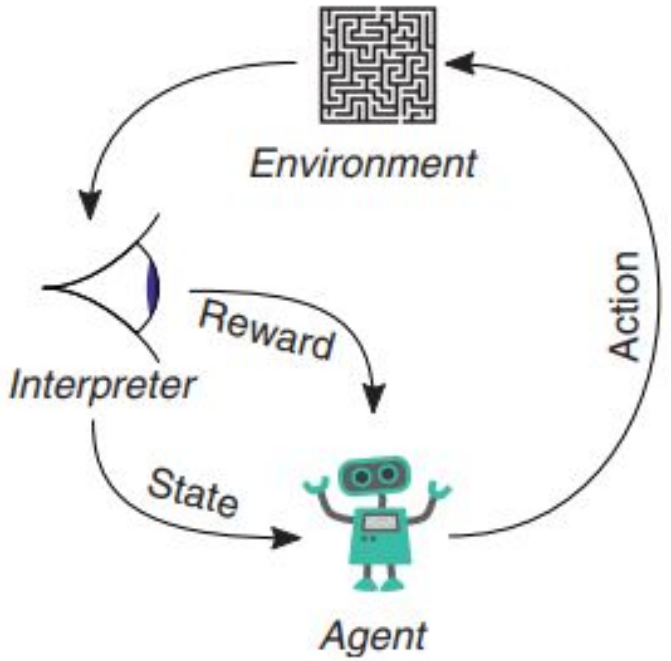
Reinforcement Learning

- attempts to model how software agents should take actions in an environment that will maximize some form of cumulative reward.
- Example : to train a computer to play chess against a human. In such a case, determining the best move to make depends on a number of factors. The number of possible states that can exist in a game is usually very large.
- To cover these many states using a standard rules-based approach would mean specifying a lot of hardcoded rules.
- RL cuts out the need to manually specify rules, and RL agents learn simply by playing the game.
- In RL, the algorithm decides to choose the next course of action once it sees a new data point.
- Based on how suitable the action is, the learning algorithm also gets some incentive a short time later.
- The algorithm always modifies its course of action toward the highest reward.
- Reinforcement learning is common in robotics, where the set of sensor readings at one point in time is a data point, and the algorithm must choose the robot's next action

Reinforcement Learning

The basic reinforcement learning (RL) model consists of the following

1. a set of environment and agent states S
2. a set of actions A of the agent
3. policies of transitioning from states to actions
4. rules that determine the scalar immediate reward of a transition and
5. rules that describe what the agent observes.



Reinforcement Learning

- An RL agent interacts with its environment in discrete time steps and in a particular order.

1. At each time t, the agent receives an observation ot, which typically includes the reward rt.

2. It then chooses an action at from the set of actions available, which is subsequently sent to the environment.

3. The environment moves to a new state st+1 from st and the reward rt+1 associated with the transition st, at, st+1 is determined.

4. The goal of a reinforcement-learning agent is to collect as much reward as possible. The agent can choose any action as a function of the history and it can even randomize its action selection

Data Collection Methods

- Surveys
- Survey Question Types
- Survey Audience
- Survey Services
- Analyzing Survey Data
- Pros and Cons of Surveys
- Interviews and Focus Groups
- Why Do an Interview?

- Why Focus Groups?
- Interview or Focus Group Procedure
- Analyzing Interview Data
- Pros and Cons of Interviews and Focus Groups
- Log and Diary Data
- User Studies in Lab and Field

Data Collection Methods

- Surveys
- four reasons to conduct a survey:
  - 1. Uncover answers
  - 2. Evoke discussion
  - 3. Base decisions on objective information
  - 4. Compare results

Data Collection Methods

Survey Question Types

- Multiple-choice-type questions
- Rank-order-type questions

  (Preferred transport – Bus, Car, Bike, Air)
- Rating or Close-ended questions

  (1 to 5)/ Strongly agree, agree, neutral, disagree, strongly disagree
- open-ended question
- Dichotomous (closed-ended) questions (Yes/No, true/flse)

Data Collection Methods

Survey Audience

who will be taking answering survey

online survey

received from right kind of audience

human took your survey/ online bots =>

same individual responds more than once

randomly clicking around your multiple-choice questions?

How all these addressed?

Survey Services

- mobile social media or old-fashioned email, piece of paper, Google Forms, instant poll, asking people to answer on the spot

Data Collection Methods

- Analyzing Survey Data
- Pros and Cons of Surveys
- People may not answer the questions honestly. You can ask them to be honest, but you can’t ask them to swear to be honest.
- You can boost the probability of honesty by telling them no one will know who answered what – that it is a blind survey and the responses will be pooled

Survey – Hypothesis : Healthcare app improves the quality of the life

1. Age

- ◯ <25 Years
- ◯ 25 - 40 Years
- ◯ >40 Years

2. The current discipline of work / Education

- ◯ Medical
- ◯ Engineering
- ◯ Allied Medical Workers
- ◯ Others

3. Do you own a smartphone/ Tablet?

- ◯ Yes
- ◯ No

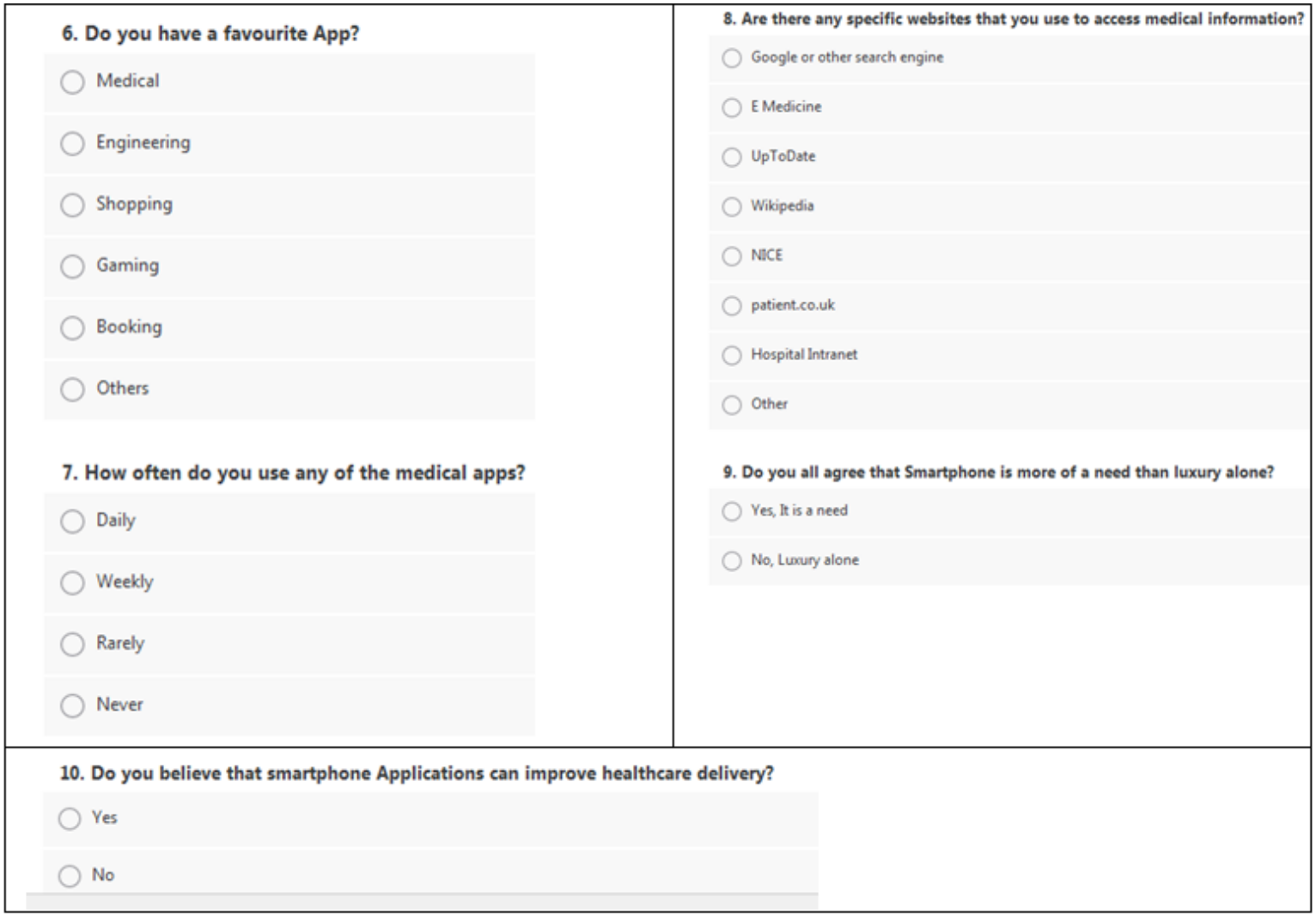
4. If so, What?

- ◯ Iphone
- ◯ Ipad
- ◯ Blackberry
- ◯ Android
- ◯ Windows
- ◯ Others

5. Do you have any medical apps?

- ◯ Yes
- ◯ No

Survey – Hypothesis : Healthcare app improves the quality of the life



6. Do you have a favourite App?

- Medical
- Engineering
- Shopping
- Gaming
- Booking
- Others

7. How often do you use any of the medical apps?

- Daily
- Weekly
- Rarely
- Never

8. Are there any specific websites that you use to access medical information?

- Google or other search engine
- E Medicine
- UpToDate
- Wikipedia
- NICE
- patient.co.uk
- Hospital Intranet
- Other

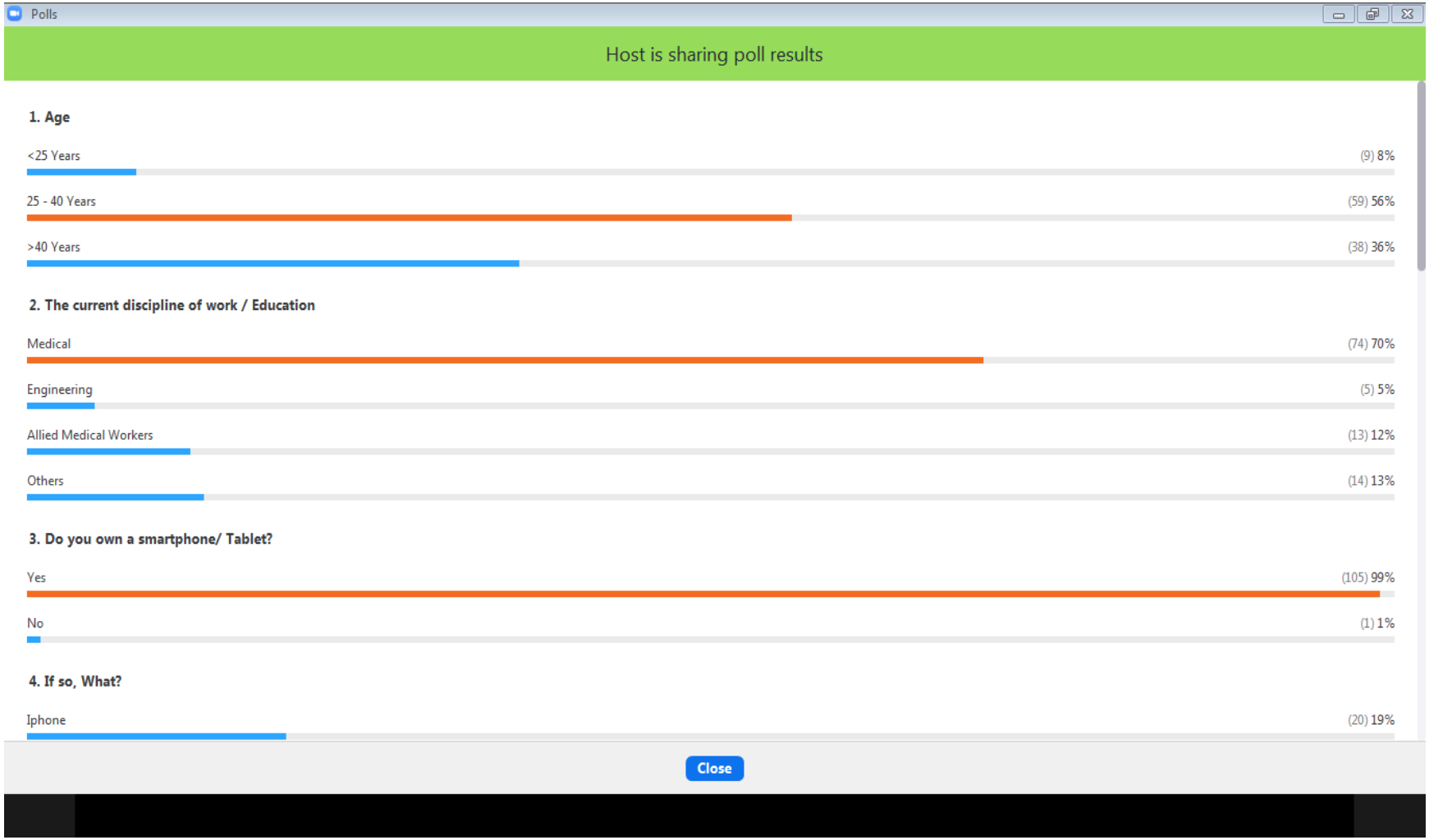
9. Do you all agree that Smartphone is more of a need than luxury alone?

- Yes, It is a need
- No, Luxury alone

10. Do you believe that smartphone Applications can improve healthcare delivery?

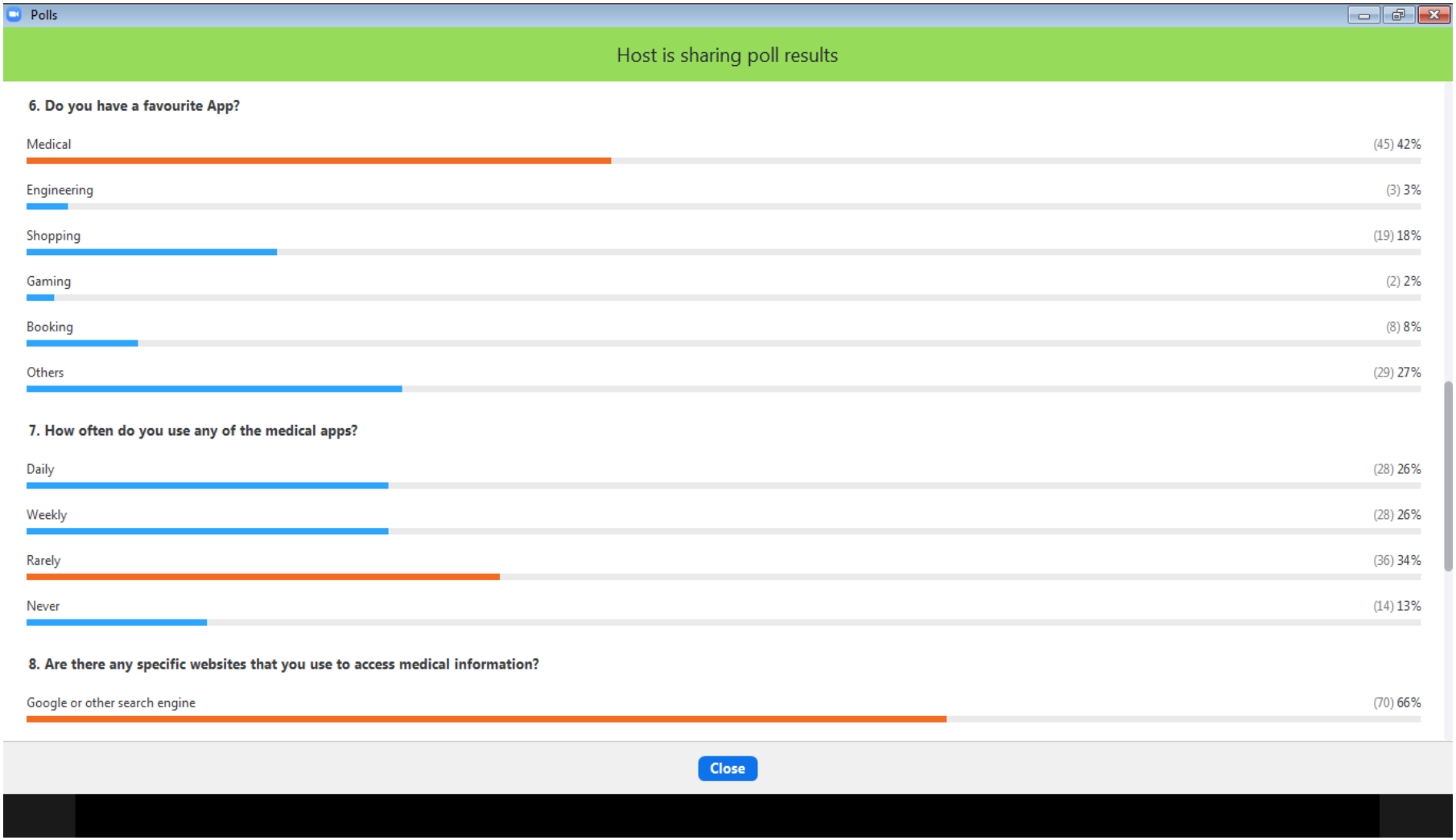
- Yes
- No

Survey – Hypothesis : Healthcare app improves the quality of the life

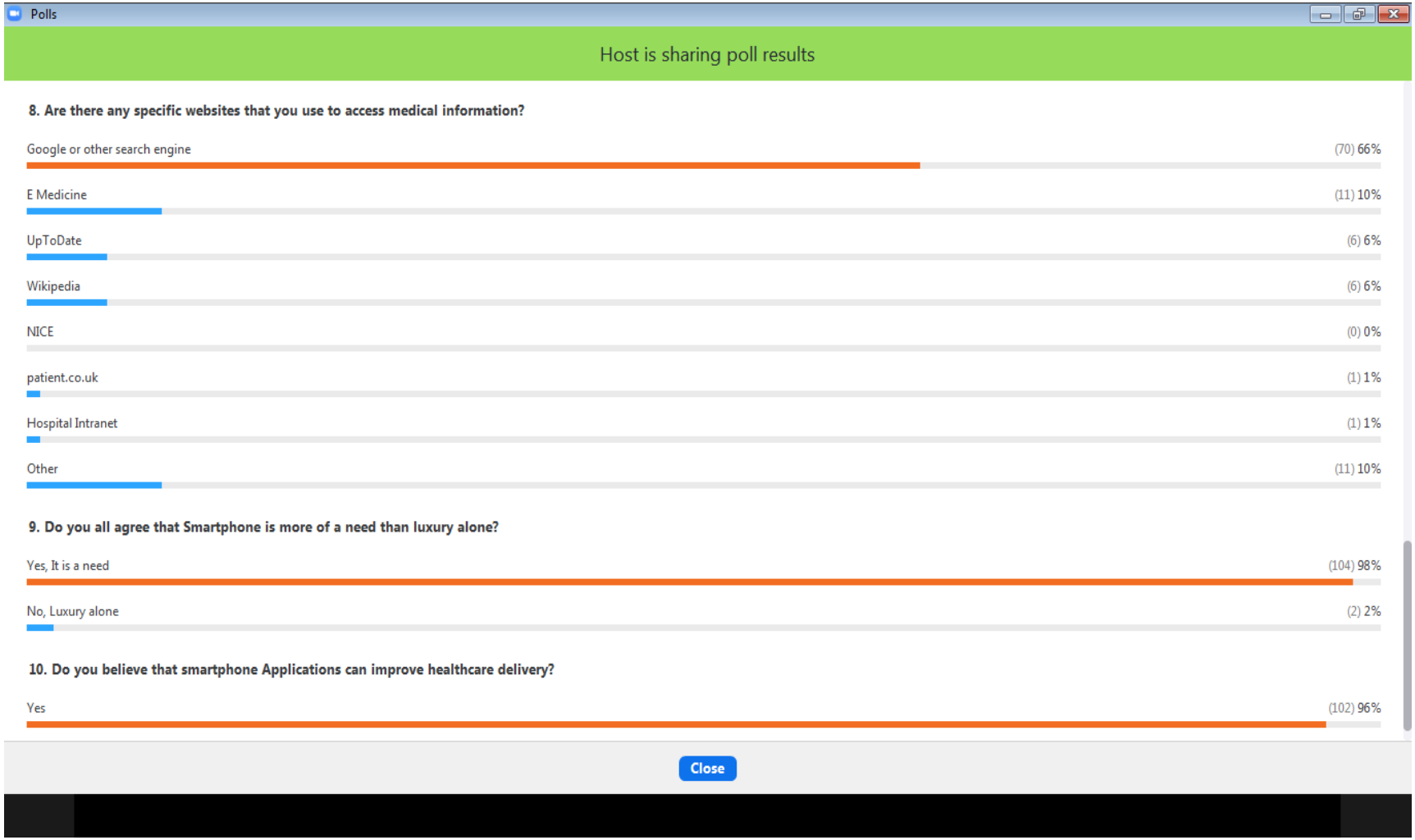


Survey – Hypothesis : Healthcare app improves the quality of the life

Survey – Hypothesis : Healthcare app improves the quality of the life



Survey – Hypothesis : Healthcare app improves the quality of the life



Data Collection Methods

- Interviews and Focus Groups

  start with

  – Agreement, Ice breaker, Honest opinion, plan
- Pros and Cons of Interviews and Focus Groups
- Log and Diary Data
- User Studies in Lab and Field

Picking Data Collection and Analysis Methods

1. Quantitative Methods
   - measuring techniques that result in numbers, or data
   - begins with an observation that leads to a hypothesis **(inductive process)**
   - done using descriptive statistics: measures of central tendency (mean, median, mode), distribution of the data (range, variance, standard deviation)
2. Qualitative Methods
   - observations of behavior, attitudes, or opinions, resulting in an assessment of qualities
   - test the hypothesis and collect numbers, information, or data, which you analyze to deduce another, perhaps closer-to-accurate hypothesis, and continue to evolve your working theory into a model **(deductive process)**
   - cannot be experimentally examined or measured

Picking Data Collection and Analysis Methods

Mixed Method Studies

Weaknesses of qualitative and quantitative methods

- Quantitative methods are often weak in understanding the context or setting in which data is collected and the results are less interpretable.
- Qualitative research, however, is time-consuming, and it may include biases from the data due to small sample sizes which may not lend itself to statistical analysis and generalization to a larger population. Mixed method strategies can offset these

Picking Data Collection and Analysis Methods

- closed-ended information => descriptive aalysis
- open-ended questions
- information retrieval
  - relationship between peoples' intentions of engaging in longer search episodes and their Web search behaviors
  - quantitative data => query logs, the amount of time spent in each query segment, query reformulation strategies
  - cluster similar search behaviors

Picking Data Collection and Analysis Methods

- Comparing Models
- True Positive (TP) is an outcome where the model correctly predicts the positive class.
- True Negative (TN) is an outcome where the model correctly predicts the negative class.
- False Positive (FP) is an outcome where the model incorrectly predicts the positive class
- False Negative (FN) is an outcome where the model incorrectly predicts the negative class
- Precision (positive predictive value)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

Picking Data Collection and Analysis Methods

- Precision (positive predictive value)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$F\text{-measure} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

Picking Data Collection and Analysis Methods

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Confusion Matrix

		Predicted condition	
	Total 8 + 4 = 12	Cancer 7	Non-cancer 5
Actual condition	Cancer 8	6	2
	Non-cancer 4	1	3

Picking Data Collection and Analysis Methods

- **Overfitting vs underfitting**
- AIC (Akaike Information Criterion):
- BIC (Bayesian Information Criterion):

- AIC and BIC attempt to solve **overfitting** problem by introducing a penalty for the number of parameters in the model
- A/B Testing
- 80/20 train/test split
- Cross-Validation
  - k-fold cross-validation
  - Leave-one-out cross-validation