Interview Query Solution #26 | HHT or HTT

Here's the question and answer for yesterdays probability question.

This question was asked by: **Goldman Sachs**

You're given a fair coin. You flip the coin until either **Heads Heads Tails** (HHT) or **Heads Tails Tails** (HTT) appears. Is one more likely to appear first? If so, which one and with what probability?

Solution:

Okay, given the two scenarios, we can assess that both sequences need H first. Once H appears, the probability of HHT is now equivalent to 1/2.

Why is this the case? Because in this scenario all you need for HHT is one H. The coin does not reset as we are flipping the coin continuously in sequence until we see the string of HHT or HTT happening in a row. Given that the first letter starts with H, this increases the chances of HHT occuring versus HTT.

Look at these scenarios where we flip the coin four times but with H showing up in the beginning each time and survey the entire sample space.

H-H-H-T = HHT
H-T-H-T = NA
H-H-T-H = HHT
H-T-H-H = NA
H-T-T-H = HTT
H-T-T-T = HTT
H-H-T-T = HHT
H-H-H-H = NA

HHT shows up more than HTT given the limited sample space of 4 flips. Increase this to 5 and it will show up even more.

The probability of HTT is 1/4 because TT needs to occur which is (1/2)*(1/2). Thus HHT is twice is likely to appear first. So, if the probability that HTT appears first is X, then the probability that HHT appears first is 2X. Since these are disjoint and together exhaust the whole probability space, X+2X=1. Therefore **X = 1/3 = HTT**

HHT is more likely to appear first than HT and the probability of HHT appearing first is 2/3.

Interview Query Data Science Answer #16 | Is it raining in Seattle?

Here's the question and answer for yesterdays probability question.

This question was asked by: Facebook

You are about to get on a plane to Seattle. You want to know if you should bring an umbrella. You call 3 random friends of yours who live there and ask each independently if it's raining. Each of your friends has a 2/3 chance of telling you the truth and a 1/3 chance of messing with you by lying. All 3 friends tell you that "Yes" it is raining.

What is the probability that it's actually raining in Seattle?

1.1 Solution:

This question can be solved in two ways in the schools of thought: Bayesian or Frequentist. The frequentist method is probably the easiest.

For example. The question prompt states, that each friend has a 2/3 change of telling the truth. Through logical transference, given that all of the friends have told you that it is raining, the question of "what is the probability that it is not raining" is the same thing as "what is the probability that all of your friends are lying?"

P(Not Raining) = P(Friend 1 Lying) AND P(Friend 2 Lying) AND P(Friend 3 Lying)

Given this logical expression. We can simply the problem to then to calculate the inverse of three AND functions. So the probability of it raining is then equated to:

P(Raining) = 1 - P(3 Friend's Lying)

Multiple of all independent probabilities:

P(3 Friend's Lying) = 1/3 * 1/3 * 1/3 = 1/27

P(Raining) = 1 - 1/27 = **26/27**

Interview Query Solution #47 | Same Side Probability

Here's the question and answer for yesterdays probability question.

This question was asked by: **Linkedin**

Suppose we have two coins. One is fair and the other biased where the probability of it coming up heads is 3/4.

Let's say we select a coin at random and flip it two times. What is the probability that both flips result in the same side?

Solution:

Let's tackle this by solving first splitting up the probabilities of getting the same side twice for the biased coin and then computing the same thing for the fair coin.

First the biased coin. We know that if we flip the biased coin we have a 3/4 chance of getting heads. And so the probability of heads twice will be **3/4 * 3/4** and the probability of tails twice is **1/4 * 1/4.**

Easy, but now what's the probability of it being either twice heads or twice tails? In this case, because the computation is an OR function, the probability is additive. In which the probabilities of **heads twice OR tails twice is computed by adding the probabilities together**.

(3/4) * (3/4) + (1/4) * (1/4) = 10/16 = **0.625**

Now the fair coin. We can apply the same formula from the biased coin to the fair coin. Since heads and tails are both equivantely probable, we can compute the formula quite easily with:

(1/2) * (1/2) + (1/2) * (1/2) = **1/2**

Now let's compute the total probability given a random selection of either coin. Since there are only two coins and we are equally likely to pick either of them, the probability of getting each is 1/2. We can then compute the total probability by again adding the individual probabilities while multiplying by the probability of choosing either.

= 1/2 * P(Biased coin same side twice) + 1/2 * P(Fair coin same side twice)
= 1/2 * (10/16) + 1/2 * (1/2) = **0.5625**

Interview Query Solution #43 I Biased five out of six

Here's the question and answer for yesterdays probability question.

This question was asked by: **Google**

Let's say we're given a biased coin that comes up heads 30% of the time when tossed.

What is the probability of the coin landing as heads exactly 5 times out of 6 tosses?

Solution:

Let's break down the prompt. We're given two pieces of information in this scenario.

1. Heads comes up 30% of the time in a biased coin.
2. What's the probability of exactly 5 heads out of 6 tosses.

Given these paramaters we can immediately infer using the **binomial distribution**!

We're given two parameters in which in a binomial distribution there exists **n independent experiments**, each asking a yes–no question, and each with its own boolean-valued outcome: success/yes/true/one with **probability p**.

$$f(k,n,p) = \Pr(k;n,p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

In this case we can assume p, the probability, is 40% from the heads-tails/yes-no question, and n, is the six tosses that we're making. The variable k is the five heads that we want to predict the probability outcome of.

Given these parameters we can now calculate the probability of by plugging in the variables. Notice that the first part of the equation represents **n choose k** which is equated to this:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Now calculating it out:

$$\Pr(5 \text{ heads}) = f(5) = \Pr(X = 5) = \binom{6}{5} 0.3^5 (1-0.3)^{6-5} = 0.010206$$

Interview Query Solution #42 | Jars and Coins

Here's the question and answer for yesterdays probability question.

This question was asked by: **Google**

A jar holds 1000 coins. Out of all of the coins, 999 are fair and one is double-sided with two heads. Picking a coin at random, you toss the coin ten times.

Given that you see 10 heads, what is the probability that the coin is double headed and the probability that the next toss of the coin is also a head?

Give your answer to 3 significant figures.

Solution:

When given the prompt it's important to think about this problem in two steps. The setting consists of 999 fair coins and one biased coin. The question itself is asking about an event probability of whether a toss will be heads. This questions should scream bayes theorem.

Let's calculate the answer by first splitting up the prior probability from the event probability.

Prior
P(fair)= .999
P(double-headed) = .001

Event Probability
Probability if fair = .5^10 = 0.0009765625
Probability if double-headed: 1

Cool, now we have both the event probabilities and the priors. Now let's define which conditions satisfy bayes theorem. We want to find the probability of having the double-sided coin given ten heads.

$$P(D \mid 10\ H) = \frac{P(10\ H \mid D) * P(D)}{P(10\ H)}$$

Let's solve for the numerator. What's the probability of 10 heads given a double headed coin? 100%. What's the probability of getting the double headed coin? 1/1000. So **P(10 H | D) * P(D) = 1/1000**

Now the denominator. The probability of 10 heads is a combination of the probability of 10 heads given a double-headed coin multiplied by the probability of picking the double-headed coin **PLUS** the probability of 10 heads given a fair coin mulitiplied by the probability of picking a fair coin.

P(10 H) = P(10 H | D) * P(D) + P(10 H | Fair) * P(Fair)

The first prior probability we already calculated in the numerator which was 1/1000. The second we can calculate P(10 H | Fair) = 0.5 ^ 10 = 0.0009765625. Probability of picking a fair coin is 999/1000. So total we have **P(10 H) = 1/1000 + (0.5^10 * 999/1000) = 0.00197558593**

Now we can finally calculate the probability of the coin being double-headed as P(D | 10 H) = .001 / 0.00197558593 = **0.506**

Given this information, we can calculate the probability of heads on the next flip the same way with bayes:

P(Heads) = P(D) * P(Heads | D) + P(Fair) * P(Heads | Fair)
= 0.506 * 1 + ((1-0.506) * 0.5) = **0.753**

Interview Query Solution #40 | Median probability

Here's the question and answer for yesterdays statistics question.

This question was asked by: **Google**

Given three random variables independent and identically distributed from a uniform distribution of 0 to 4, what is the probability that the median is greater than 3?

Solution:

If we break down this question, we'll find that another way to phrase it is to ask what the probability is that **at least two of the variables are larger than 3.**

For example, if look at the combination of events that satisfy the condition, the events can actually be divided into two exclusive events.

Event A: All three random variables are larger than 3.
Event B: One random variable is smaller than 3 and two are larger than 3.

Given these two events satisfy the condition of the median > 3, we can now calculate the probability of both of the events occuring. The question can now be rephrased as **P(Median > 3) = P(A) + P(B).**

Let's calculate the probability of the event A. The probability that a random variable > 3 but less than 4 is equal to 1/4. So the probability of event A is:

P(A) = (1/4) * (1/4) * (1/4) = 1/64

The probability of event B is that two values must be greater than 3, but one random variable is smaller than 3. We can calculate this the same way as the calculating the probability of A. The probability of a value being greater than 3 is 1/4 and the probability of a value being less than 3 is 3/4. Given this has to occur three times we multiply the condition three times.

P(B) = 3 * ((3/4) * (1/4) * (1/4)) = 9/64

Therefore the total probability is P(A)+P(B) = 1/64 + 9/64 = **10/64**

Interview Query Solution #39 | Biased random number generator

---

Here's the question and answer for yesterdays probability question.

This question was asked by: **Amazon**

Given an unfair coin with the probability of heads and tails not equal to 50/50, what algorithm could generate a list of random ones and zeros?

Solution:

This problem can be solved with a method called the von neumann corrector. Observe that even if the probability is not 50/50, we can get an equal distribution of two values by taking the combination of outputs.

The algorithm works on pairs of bits, and produces output as follows:
1. When you get heads-tails you count the toss as heads or 1.
2. When you get tails-heads you count it as tails or zero.
3. You ignore the throws that come up twice the same side whether it's TT or HH.

Regardless of the distribution of heads and tails, the output will always have a 50/50 split of 0s and 1s. The algorithm will discard (on average) 75% of all inputs however, even if the original input was perfectly random to start with.

Interview Query Solution #29 | Rejection Reason

Here's the question and answer for yesterdays modeling question.

This question was asked by: **Affirm**

Suppose we have a binary classification model that classifies whether or not an applicant should be qualified to get a loan. Because we are a financial company we have to provide each rejected applicant with a reason why.

Given we only have access to the feature weights, how would we give each rejected applicant a reason why they got rejected?

Solution:

Let's pretend that we have three people: Alice, Bob, and Candace that have all applied for a loan. Simplifying the financial lending loan model, let's assume the only features are **total number of credit cards**, **dollar amount of current debt**, and **credit age**.

Let's say Alice, Bob, and Candace all have the same number of credit cards and credit age but not the same dollar amount of curent debt.

Alice: 10 credit cards, 5 years of credit age, **$20K** of debt
Bob: 10 credit cards, 5 years of credit age, **$15K** of debt
Candace: 10 credit cards, 5 years of credit age, **$10K** of debt

Alice and Bob get rejected for a loan but **Candace gets approved**. We would assume that given this scenario, we can logically point to the fact that Candace's 10K of debt has swung the model to approve her for a loan.

How did we reason this out? If the sample size analyzed was instead thousands of people who had dthe same number of credit cards and credit age with varying levels of debt, we could figure out the model's average loan acceptance rate for each numerical amount of current debt. Then we could plot these on a graph to **model out the y-value, average loan acceptance, versus the x-value, dollar amount of current debt**.

These graphs are called **partial depedence plots**!

The partial dependence plot is calculated only after the model has been fit. The model is fit on the real data. In that real data, loans are given dependent on different feaures. But after the model is fit, we could start by taking all the characteristics of a loan and plotting them against the dependent variable **whilst keeping all of the other features the same** except for the one feature variable we want to test.

We then use the model to predict the loan qualification but we change the debt dollar amount before making a prediction. We first predict the loan qualification for an example person by setting it to 20K. We then predict it again at $19K. Then predict again for $18K. And so on. We trace out how predicted probability of loan qualification (on the vertical axis) as we move from small values of debt dollar amount to large values (on the horizontal axis). This way we are able to see how a model's features affect the score without digging into the classifier feature weights.

deeplearning.ai

THE BATCH

September 18, 2019 Essential news for deep learner

Subscribe Tips

Dear friends,

Over the weekend, we hosted our first Pie & AI meetup in Kuala Lumpur, Malaysia, in collaboration with the AI Malaysia group, MDEC, and ADAX. The event was part of Malaysia's AI & Data Week 2019. Several people traveled from neighboring southeast Asian countries to attend!

I'm glad to see so many AI communities growing around the world, and I'm excited to bring more exposure to them. If you'd like to partner with us for a Pie & AI event, I hope you'll drop us a note at events@deeplearning.ai.

Keep learning!

Andrew

News

Agbots Want Jobs Americans Don't

Advances in computer vision and robotic dexterity may reach the field just in time to save U.S. agriculture from a looming labor shortage.

What happened: *CNN Business* surveyed the latest crop of AI-powered farmbots, highlighting those capable of picking tender produce, working long hours, and withstanding outdoor conditions.

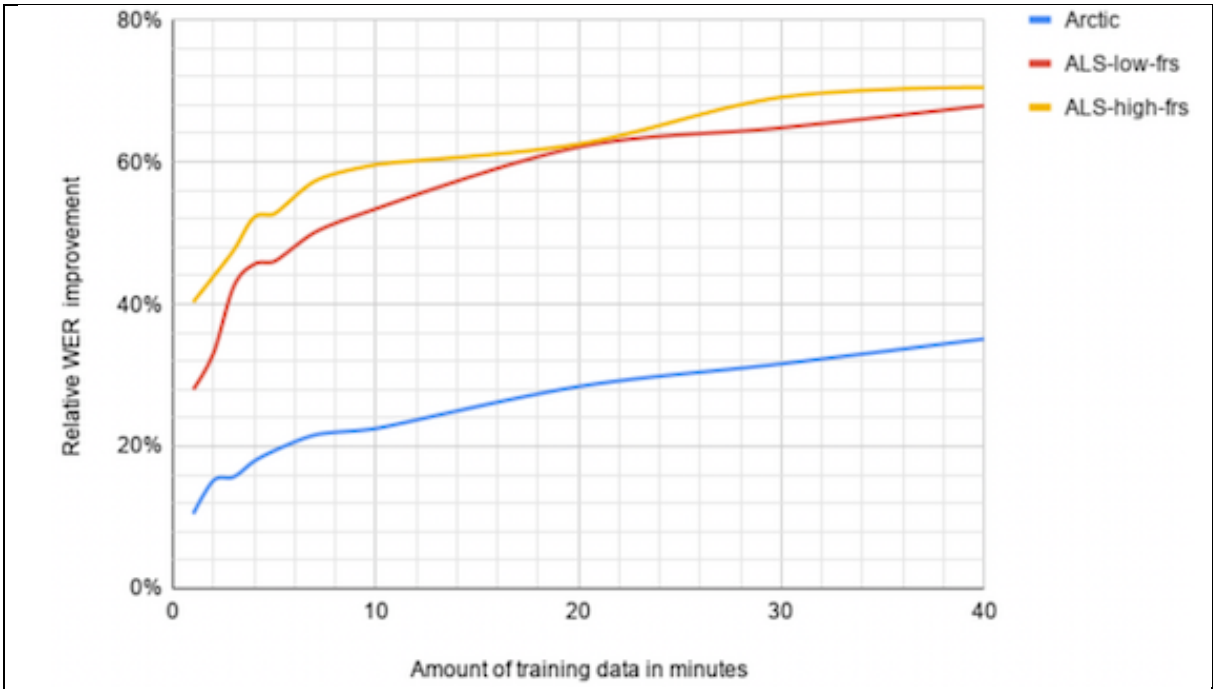
Robot field hands: Harvest bots tend to use two types of computer vision: one to identify ripe fruits or vegetables, the other to guide the picker.

- Vegebot, a lettuce harvester developed at the University of Cambridge, spots healthy, mature heads of lettuce with 91 percent accuracy and slices them into a basket using a blade powered by compressed air. The prototype harvests a head in 30 seconds, compared to a human's 10-second average. The inventors say with lighter materials, they could catch up.
- Agrobot's strawberry-picking tricycle straddles three rows of plants. It plucks fragile berries using up to 24 mechanical hands, each equipped with a camera that grades the fruit for ripeness.
- California's Abundant Robotics built a rugged, all-weather autonomous tractor that vacuums up ripe apples (pictured above).

Behind the news: Unauthorized migrants do as much as 70 percent of U.S. harvest work, according to a study by the American Farm Bureau Association. Tighter immigration policies and improving opportunities at home increasingly keep such workers out of the country.

Why it matters: The shortage of agricultural workers extends across North America. During harvest season, that means good produce is left to rot in the fields. The situation costs farmers millions in revenue and drives up food prices.

Our take: The robots-are-coming-for-your-job narrative often focuses on people put out of work but fails to acknowledge that workers aren't always available. Between a swelling human population and emerging challenges brought on by climate change, the agriculture industry needs reliable labor more than ever. In some cases, that could be a machine.



Speech Recognition With an Accent

Models that achieve state-of-the-art performance in automatic speech recognition (ASR) often perform poorly on nonstandard speech. New research offers methods to make ASR more useful to users with heavy accents or speech impairment.

What's new: Researchers at Google fine-tuned ASR neural networks on a data set of heavily accented speakers, and separately on a data set of speakers with amyotrophic lateral sclerosis (ALS), which causes slurred speech of varying degree. Their analysis shows marked improvement in model performance. The remaining errors are consistent with those associated with typical speech.

Key insight: Fine-tuning a small number of layers closest to the input of an ASR network produces good performance in atypical populations. This contrasts with typical transfer learning scenarios, where test and training data are similar but output labels differ. In those scenarios, learning proceeds by fine-tuning layers closest to the output.

How it works: Joel Shor and colleagues used data from the L2-ARCTIC data set for accented speech and ALS speaker data from the ALS Therapy Development Institute. They experimented with two pre-trained neural models, RNN-Transducer (RNN-T) and Listen-Attend-Spell (LAS).

- The authors fine-tuned both models on the two data sets with relatively modest resources (four GPUs over four hours). They measured test-set performance on varying amounts of new data.
- They compared the sources of error in the fine-tuned models against models trained on typical speech only.

Results: RNN-T achieved lower word error rates than LAS, and both substantially outperformed the Google Cloud ASR model for severe slurring and heavily accented speech. (The three models were closer with respect to mild slurring, though RNN-T held its edge.) Fine-tuning on 15 minutes of speech for accents and 10 minutes for ALS brought 70 to 80 percent of the improvement.

Why it matters: The ability to understand and act upon data from atypical users is essential to making the benefits of AI available to all.

Takeaway: With reasonable resources and additional data, existing state-of-the-art ASR models can be adapted fairly easily for atypical users. Whether transfer learning can be used to adapt other types of models for broader accessibility is an open question.

Generative Models Rock

AI's creative potential is becoming established in the visual arts. Now musicians are tapping neural networks for funkier grooves, tastier licks, and novel subject matter.

What happened: Aaron Ackerson, whom the *Chicago Sun-Times*called "a cross between Beck and Frank Zappa," produced his latest release with help from the latest generation of generative AI. MuseNethelped generate the music and GPT-2 suggested lyrics. DeepAI's Text To Image API synthesized the cover art.

Making the music: “Covered in Cold Feet” began its existence as an instrumental fragment scored for violin, piano, and bass guitar.

- Ackerson fed a two-bar MIDI file into MuseNet, which spat out a few more bars based on his raw material.
- He tweaked MuseNet’s output to his liking using his digital audio workstation. Then he fed that material back to MuseNet, which expanded by a few bars more, repeating the process until he had a composition he was happy with.
- “Early in the process, I had MuseNet generate continuations of my simple idea in a lot of different styles, and most did not find their way into the finished song,” Ackerson said in an interview with *The Batch*. “The one I decided to use followed the main melody with what would later turn into the beginning of the guitar solo.”
- That solo combusts into righteous shredding near the 1:25 mark, a triumph of manual dexterity as much as AI.

Writing the lyrics: The groove reminded Ackerson of the band Phish. So he fed a list of that band's song titles to Talk to Transformer, an online text-completion app based on the half-size version of GPT-2.

- "Covered in Cold Feet" was his favorite of its responses to the original list.
- He repeatedly fed Talk to Transformer the phrase "covered in cold feet again," and curated the lyrics from its responses.
- Talk to Transformer doesn't generate rhymes, so Ackerson added them manually.

Behind the music: The artist composed his first AI-assisted song in 2017 using the Botnik Voicebox text generator. He fed the model bass and melody lines from 100 of his favorite songs translated into solfège (a note-naming system that maps the tones in any musical key to the syllables do, re, mi, and so on). The model spat out fresh note pairings, many of which he had never before considered using. The result, "Victory Algorithm," is a slide guitar-fueled psychobilly foot stomper.

We're thinking: AI skeptics worry that computers, if allowed to do creative work, will erase humanity from art. No worries on that point: Ackerman's personality comes through loud and clear. We look forward to hearing more from musicians brave enough to let computers expand their creative horizons. (For more on MuseNet, see our interview with project lead Christine Payne.)

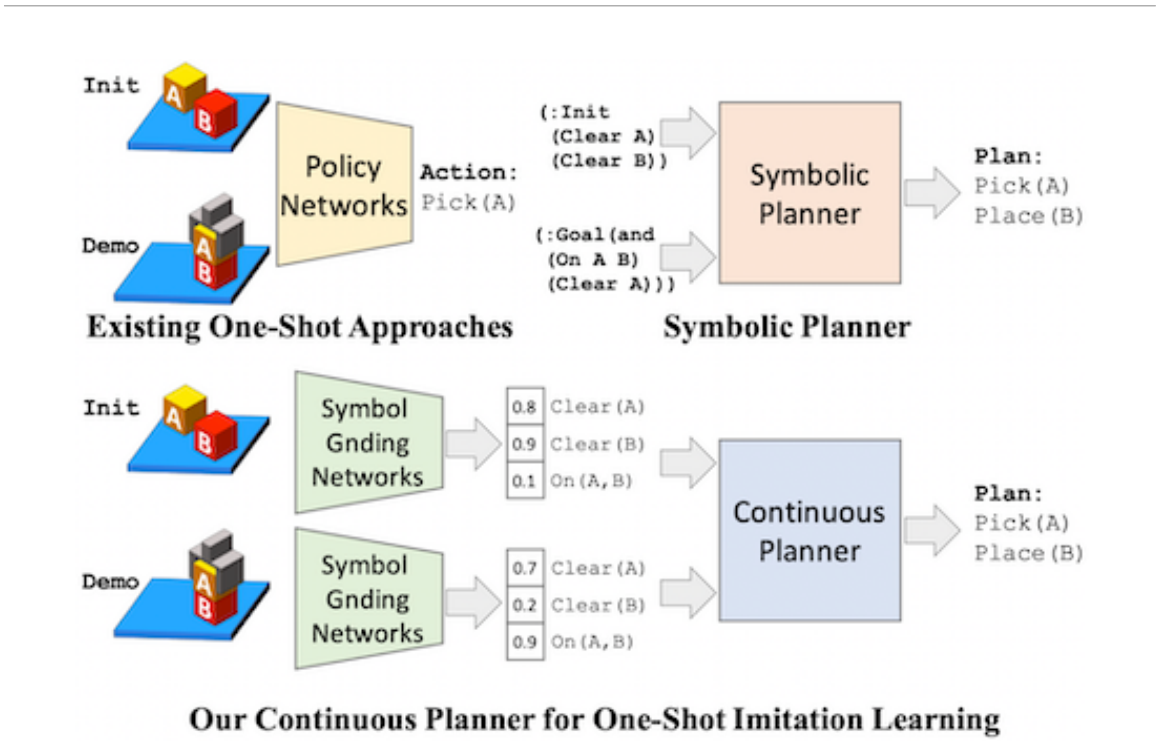
My Chatbot Will Call Your Chatbot

Companies with large numbers of contractual relationships may leave millions of dollars on the table because it's not practical to customize each agreement. A new startup offers a chatbot designed to claw back that money.

What happened: Pactum, a startup that automates basic vendor and service contracts at immense scale, emerged from stealth with a $1.15 million investment from Estonian tech upstart Jaan Tallinn and his posse of Skype alumni.
How it works: Let's say a prominent computer company develops a new laptop and hires Pactum to cut distribution deals with hundreds of thousands of computer stores around the globe.

- Pactum's AI model reviews the computer maker's existing contracts to establish baseline terms.
- Then it examines variables such as pricing, schedule, and penalties in search of more favorable arrangements. For instance, it may seek to improve cash flow by asking retailers to pay for orders faster.
- The AI then initiates negotiations via chatbot.
- The model automatically updates contract terms as negotiations proceed.

Behind the news: Contracts are a hot area for AI. In 2015, Synergist.ioand Clause launched automated platforms that mediate contract negotiations. And last year, Dutch information services firm Wolters Kluwer acquired legal AI startups CLM Matrix and Legisway.
Why it matters: Standardized contracts can save time and effort spent customizing agreements. But they also bring costs. A 2018 study by KPMG estimated that standard contracts can soak up between 17 and 40 percent of a contract's expected revenue.
The Tallinn Effect: Funding from Jaan Tallinn brings the credibility of a serial entrepreneur who co-founded Skype and Kazaa and invested in DeepMind. It's also a stamp of approval from a technologist who thinks deeply about AI's potential for both benefit and harm. Tallinn co-founded the Centre for the Study of Existential Risk and once wrote, "In a situation where we might hand off the control to machines, it's something that we need to get right." Apparently he believes Pactum meets that standard.



Working Through Uncertainty

How to build robots that respond to novel situations? When prior experience is limited, enabling a model to describe its uncertainty can enable it to explore more avenues to success.

What's new: In reinforcement learning, meta-learning describes teaching a model how to complete multiple tasks, including tasks the model hasn't seen before. One way to approach meta-learning is to divide it into two subproblems: creating a plan based on current surroundings and the task at hand, and taking action to implement the plan. Stanford researchers developed deep learning models that facilitate the planning phase by learning to generate better representations of the task.

Key insight: Deep learning has been used to learn vector descriptions of the initial state prior to accomplishing a task and the final state afterward. The new work uses probabilistic descriptions, allowing more flexibility in novel tasks. For example, instead of having to choose between the contradictory descriptions *object 1 is on object 2* and *object 2 is on object 1*, the network updates its confidence in each statement throughout the planning steps.

How it works: Previous methods use a neural network model as a classifier to decide state descriptions from potential configurations. Instead, De-An Huang and his colleagues use the model's confidence in each potential configuration to represent states. This approach produces a probabilistic description of current and final states.

- For training, the model takes a set of demonstrations of similar tasks plus the actions available to the planning algorithm. For testing, it takes a single demonstration of a novel task, the initial state, and the allowed operations.
- For both initial and final states, a network is trained to predict the probability that certain configurations are observed. For example, *based on an image, learn the probability that object 1 is on top of object 2.*
- The planning algorithm takes the probabilistic descriptions and selects the action most likely to move the initial state closer to the final state. Since the choice is a function of the state descriptions and potential operations, the planning algorithm requires no training.

Results: The authors' approach achieves state-of-the-art meta-learning performance in sorting objects and stacking blocks. When sorting, it matches performance based on human heuristics. When stacking, it outperforms human heuristics plus fixed state descriptions with less than 20 training examples (although the heuristics win with 30 training examples).

Yes, but: The researchers achieved these results in tasks with a small number of operations and potential state configurations. Their method likely will struggle with more complex tasks such as the Atari games that made deep reinforcement learning popular.

Takeaway: In past models, misjudgments of surroundings and goals tend to accumulate, leading the models far from the intended behavior. Now, they can relax their fixed state descriptions by representing potential points of confusion as probabilities. This will enable them to behave more gracefully even with little past experience to draw on.

Easing Cross-Border Collaboration

Tighter national borders impede progress in AI. So industry leaders are calling for a different kind of immigration reform.

What happened: The Partnership on AI published a report calling out the corrosive effect of restrictive immigration policies and suggesting alternatives.

What it says: The nonprofit industry consortium's report calls for countries aiming to grow their AI industry to ease visa requirements for high-skilled tech workers. Among its recommendations:

- Countries should present visa rules clearly and make the review process transparent, so applicants know they are being judged on skill. Get rid of caps, too.
- Provide assistance to students and low-income applicants in navigating the visa process. Countries serious about fostering AI should provide students with pathways to citizenship.
- Update immigration laws to redefine the concept of family to include long-term partnerships and nontraditional marriages.
- No nation should bar tech workers based on nationality. Moreover, countries should apply patent law, not immigration policy, to protect intellectual property.

Why it matters: Strict immigration policies limit opportunities for researchers and practitioners. They also shut out students, small businesses, startups, and small colleges from the broader AI community. In the U.S., tighter borders are cutting into the country's competitiveness in AI, according to a new study from Georgetown University's Center for Security and Emerging Technology. CSET recommends eliminating existing U.S. barriers to recruiting and retaining foreign-born AI talent.

We're thinking: Human capital is critical to AI, and the global community has an interest in channeling it toward worthwhile projects. The Partnership on AI's recommendations offer a solid foundation for international trade groups, such as the Organization for Economic Cooperation and Development, to build policies that accelerate progress by opening the world to workers in emerging tech.

Interview Query Question [Premium] #26 | HHT or HTT

Good morning. Here's your probability question for today.

This question was asked by: **Goldman Sachs**

You're given a fair coin. You flip the coin until either **Heads Heads Tails** (HHT) or **Heads Tails Tails** (HTT) appears. Is one more likely to appear first? If so, which one and with what probability?

Hi! My name is Jay and I'm one of the co-founders of Interview Query. By signing up you've completed the first step to getting your dream data job whether you're just starting out as a new graduate out of college or if you're a seasoned machine learning and data scientist pro!

Here's how Interview Query works and how you can make the most of it:

Every Monday, Wednesday, and Friday we will be sending you a data science interview question sourced from one of many top tech companies. We've tailored each question so that it should take under one hour to solve. Given the wide-breadth of topics within data science and machine learning, the question will usually be one of eight different topics:

- SQL coding
- Product intuition and problem solving
- Python scripting
- Modeling knowledge
- Machine learning concepts
- Statistics
- Probability
- Algorithms in python

Make sure to really sit down and solve the question each day for full practice. If you find the questions useful, be sure to either share your questions with friends or sign up for premium to get the full answers the next day!

Please feel free to send us any feedback or suggestions! We're really passionate about making this product useful for your needs. You can reach us anytime at this email. And definitely let us know if we helped you get a job. We love to hear all of our success stories!

Here's your first question with the complimentary solution attached.

Interview Query Data Science Question | Comments Histogram
This question was asked by: Facebook

`users` table

columns	type
id	int
name	varchar
joined_at	datetime
city_id	int
device	int

`user_comments` table

columns	type
user_id	int
body	text
created_at	datetime

Write a SQL query to create a histogram of number of comments per user in the month of January 2019. Assume bin buckets class intervals of one.

1.2 Solution:

Let's break down the solution. Here are the things we have to note.

A histogram with bin buckets of one means that we can avoid the logical overhead of grouping frequencies into specific intervals when we run a GROUP BY in SQL. Since a histogram is just a display of frequencies of each user, all we need to do is get the total count of user comments in the month of January 2019 and then group by that count.

At first look it seems like we don't need to do a join between the two tables. The `user_id` column exists in both tables. But if we ran a GROUP BY on just `user_id` on the `user_comments` table it would return the number of commments for each user right? But what happens when a user does not make a comment?

Because we still need to account for users that did not make any comments in January 2019, we need to do a left join on `users` to `user_comments`, and then take the COUNT of a field in the `user_comments` table. That way we can get a 0 value for any users that did not end up commenting in January 2019.

```
SELECT users.id, COUNT(user_comments.user_id) AS comment_count
FROM users
LEFT JOIN user_comments
   ON users.id = user_comments.user_id
WHERE created_at BETWEEN '2019-01-01' AND '2019-01-31'
GROUP BY 1
```

The above CTE gives us a table with each user id and their corresponding comment count for the month of January 2019. Now that we have the comment count for each user, all we need to do is to group by the comment count to get a histogram.

```
WITH hist AS (
  SELECT users.id, COUNT(user_comments.user_id) AS comment_count
  FROM users
  LEFT JOIN user_comments
    ON users.id = user_comments.user_id
  WHERE created_at BETWEEN '2019-01-01' AND '2019-01-31'
  GROUP BY 1
)

SELECT comment_count, COUNT(*) AS frequency
FROM hist
GROUP BY 1
```

Interview Query Data Science Question [Premium] #16 | Is it raining in Seattle?

Good morning. Here's your probability question for today.

This question was asked by: Facebook

You are about to get on a plane to Seattle. You want to know if you should bring an umbrella. You call 3 random friends of yours who live there and ask each independently if it's raining. Each of your friends has a 2/3 chance of telling you the truth and a 1/3 chance of messing with you by lying. All 3 friends tell you that "Yes" it is raining.

What is the probability that it's actually raining in Seattle?

Interview Query Data Science Answer #12 | Four Person Elevator

Here's the question and answer for yesterdays probability question.

This question was asked by: Postmates

There are four people on the ground floor of a building that has five levels not including the ground floor. They all get into the same elevator.

If each person is equally likely to get on any floor and they leave independently of each other, what is the probability that no two passengers will get off at the same floor?

1.3 Solution:

The number of ways to assigning five floors to four different people is to get the total sample space. In this case it would be 5 * 5 * 5 * 5. For each person, they can choose one of five floors, which happens four times for four people. So the total number of combinations is **5^4**.

The number of ways to assign five floors to four people without repetition of floors is 5 * 4 * 3 * 2 because for the first passenger you have five different options. The second person has four, and so on. Note that this number counts all possible orders betwen passengers as well.

The result is then 5/5 * 4/5 * 3/5 * 2/5 = **0.192**

Interview Query Data Science Answer #10 | First to Six

Here's the question and answer for yesterdays probability question.

This question was asked by: Microsoft

Amy and Brad take turns in rolling a fair six-sided die. Whoever rolls a "6" first wins the game. Amy starts by rolling first.

What's the probability that Amy wins?

1.4 Solution:

Let's set some definitions.

pA = Probability that Amy wins
pB = Probability that Brad wins.

Note that ***pA*** *= P[win if go first]*.

So we can then deduce that Brad's probability of winning then becomes the probability of going first after Amy loses the first rol. We can represent that with this equation of: *pB = P[Amy loses first roll] * P[win if go first]*.

We also know that the probabilities of either Amy or Brad winning should add up to 1. So mathematically we can create two equations: **pB = 5/6 * pA**and **pA + pB = 1**.

This is now a linear algebra question. Two equations and two unknowns.

*pA = pB - 1 --> pB = 5/6 * (pB - 1)*
*pB = 5/6pB - 5/6 --> 5/6 = 11/6pB --> pB = 5/6 * 11/6* = 5/11

The answer is then **pA = 1 - 5/11 --> 6/11**

Interview Query Data Science Answer #7 | Random Number

Here's the question and answer for yesterdays algorithm question.

This question was asked by: Facebook

Given a stream of numbers, select a random number from the stream, with O(1) space.

1.5 Solution:

We need to prove that every element is picked with 1/n probability where n is the number of items seen so far. For every new stream item x, we pick a random number from 0 to the count-1. If the picked number is count-1, we replace the previous result with x.

```
import random

# A function to randomly select a item
# from stream[0], stream[1], .. stream[i-1]
def selectRandom(x):
    res = 0
    # count of numbers visited so far in stream
    count = 0
    # increment count of numbers seen so far
    count += 1
    # if this is the first element from stream, return it
    if (count == 1):
        res = x
    else:
        # generate a random number from 0 to count - 1
        i = random.randrange(count)

        # replace the prev random number with new number with 1/count
        if (i == count - 1):
            res = x
    return res

# Driver Code
stream = [1, 2, 3, 4];
n = len(stream);

# Use a different seed value
# for every run.
for i in range (n):
    print("Random number from first",
        (i + 1), "numbers is",
```

```
        selectRandom(stream[i]));
```

To simplify proof, let us first consider the last element, the last element replaces the previously stored result with 1/n probability. So probability of getting last element as result is 1/n.

Let us now talk about second last element. When second last element processed first time, the probability that it replaced the previous result is 1/(n-1). The probability that previous result stays when nth item is considered is (n-1)/n. So probability that the second last element is picked in last iteration is [1/(n-1)] * [(n-1)/n] which is 1/n.

Similarly, we can prove for third last element and others.

Interview Query Data Science Answer #1 | Weekly Aggregation

Here's the question and answer for yesterdays python question.

This question was asked by: **Postmates**

Given a list of timestamps in sequential order, return a list of lists grouped by week (7 days) using the first timestamp as the starting point.

Example:

```
ts = [
   '2019-01-01',
   '2019-01-02',
   '2019-01-08',
   '2019-02-01',
   '2019-02-05',
]

output = [
   ['2019-01-01', '2019-01-02'],
   ['2019-01-08'],
   ['2019-02-01', '2019-02-05'],
]
```

1.6 Solution:

This question sounds like it should be a SQL question doesn't it? Weekly aggregation implies a form of GROUP BY in a regular SQL or pandas question. In either case, aggregation on a dataset of this form by week would be pretty trivial.

But since it's a scripting question, it's trying to pry out if the candidate deal with unstructured data. Data scientists deal with a lot of unstructured data.

In this function we have to do a few things.
1. Loop through all of the datetimes
2. Set a beginning timestamp as our reference point.
3. Check if the next time in the array is more than 7 days ahead.
   a. If so, set the new timestamp as the reference point.
   b. If not, continue to loop through and append the last value.

```
from datetime import datetime
nts = [] #convert to datetime for testing
for t in ts:
        nts.append(datetime.strptime(t, '%Y-%m-%d'))

def weekly_agg(ts):
        wk = []
        start_index = 0 #this will be the starting datetime index
        temp = [ts[0]]
        for i in range(1, len(ts)):
                if (ts[i] - ts[start_index]).days < 7:
                        temp.append(ts[i])
                else: # break out of loop condition
                        wk.append(temp)
                        temp = [ts[i]] #reset sub-array
                        start_index = i #reset index
        wk.append(temp)
        return wk
```

Interview Query Data Science Answer #2 | Decreasing Comments

Here's the question and answer for yesterdays product question.

This question was asked by: **Pinterest**

Let's say you work for a social media company that has just done a launch in a new city. Looking at weekly metrics, you see a slow decrease in the average number of comments per user from January to March in this city.

The company has been consistently growing new users in the city from January to March.

What are some reasons on why the average number of comments per user would be decreasing and what metrics would you look into?

1.7 Solution:

Let's take an approach of investigating into a couple of different metrics. Many candidates like to randomly shoot in the dark and think about external factors. (Maybe there's a

winter storm in Chicago and no one has power). Almost 100% of the time the answer is not related to external factors.

Average comments per user is calculated by taking the total number of comments divided by the total number of users. So let's model out an example scenario.

Jan: 10000 users, 30000 comments, 3 comments/user
Feb: 20000 users, 50000 comments, 2.5 comments/user
Mar: 30000 users, 60000 comments, 2 comments/user

We're given information that total user count is increasing linearally which means that the decreasing comments/user is not an effect of a declining user base creating a loss of network effects on the platform.

With this we can hypothesis a couple of answers:

1. Could more users cause less individual engagement?

Otherwise known as the crowding effect, we can model this by looking at a cohort of users that started in the first week of January and then another cohort of users that started the first week of March. If we see no difference between the number of comments after the first, second, third weeks, then there likely is no crowding effect happening.

2. What about active user engagement?

We know that even as the company is increasing users, it's likely users will fall off and churn off the platform. Let's say we model the user churn by month.

Month 1: 25% Churn
Month 2: 20% Churn
Month 3: 15% Churn

This means that for the cohort of users that starts in January, by February there are now only 7500 (10000 * 75%) active users, then in March 6000 active users from a 20% churn.

This can explain a likely effect of why we see a decrease of comments per user even though the total user count is increasing linearly. Churn decreases the active user counts which is assumed to be directly correlated to how many comments will exist on a platform. Because we have **less active users on the platform, the denominator is in this case a fake proxy for actual platform engagement**.

If we wanted to measure if active users will still commenting, we could then just look at **comments per active user**.

Interview Query Data Science Answer #3 | Employee Salaries

Here's the question and answer for yesterdays sql question.

This question was asked by: **Microsoft**

employees table

columns	types
id	int
first_name	varchar
last_name	varchar
salary	int
department_id	int

departments table

columns	types
id	int
name	varchar

1. Given the tables above, select the top 3 departments by the highest percentage of employees making over 100K in salary and have at least 10 employees.

Example output:

> 100K %	department name	number of employees
90%	engineering	25
50%	marketing	50
12%	sales	12

2. Let's say due to an ETL error, the employee table instead of updating the salaries every year when doing compensation adjustments, did an insert instead. The head of HR still needs the current salary of each employee. Write a query to get the current salary for each employee.

Assume no duplicate combination of first and last names. (I.E. No two John Smiths)

1.8 Solution:

1. We definitely need a JOIN, a HAVING clause, and a CASE WHEN to differentiate when a employee makes over 100K.

If we first do a join, we will get all employees and their departments. Then all that's left is grouping by the department name and calculating

- a function to get all of the employees
- a function to get all employees making over 100K
- dividing those values by each other

```
SELECT
   d.name
   , CAST(SUM(CASE WHEN salary > 100000 THEN 1 ELSE 0 END) AS DECIMAL)/COUNT(*)
AS percent_employees_over_100K
FROM departments AS d
LEFT JOIN employees AS e
   ON d.id = e.department_id
GROUP BY 1
HAVING COUNT(*) >= 10
ORDER BY 2 DESC
LIMIT 3
```

2. The first step would be to remove duplicates. Given we know there aren't any duplicate first and last name combinations, we can remove duplicates from the employees table by just grouping by first and last name and getting the maximum id from the table which would be the last entry and the most up to date salary.

This way we can rejoin by doing a subquery and not get duplicates in the existing table.

```
SELECT e.first_name, e.last_name, e.salary
FROM employees AS e
INNER JOIN (
   SELECT first_name, last_name, MAX(id) AS max_id
   FROM employees
   GROUP BY 1,2
) AS m
   ON e.id = m.max_id
```

Interview Query Data Science Answer #4 | 500 Cards

Here's the question and answer for yesterdays probability question.

This question was asked by: **LinkedIn**

Imagine a deck of 500 cards numbered from 1 to 500. If all the cards are shuffled randomly and you are asked to pick three cards, one at a time, what's the probability of each subsequent card being larger than the previous drawn card?

1.9 Solution:

Imagine this as a sample space problem ignoring all other distracting details. If you have to draw three different numbered cards without replacement, and they are all unique, then

we are assuming that there will be effectively a lowest card, a middle card, and a high card.

Let's make it easy and assume we drew the numbers 1,2, and 3. In our scenario, if we drew (1,2,3), then that would be the winning scenario. But what's the full range of outcomes we could draw? Let's map out all of the possibilities.

(3,2,1)
(3,1,2)
(2,1,3)
(2,3,1)
(1,3,2)
(1,2,3)

So six possibilities in the total sample space. And only one of them is the partition that we want. Given this, the answer is 1/6.

The trick is to not be distracted by the size of the population. The population does not matter if you are looking at the order within the sample.