StackOverflow: Can I actually use this answer?

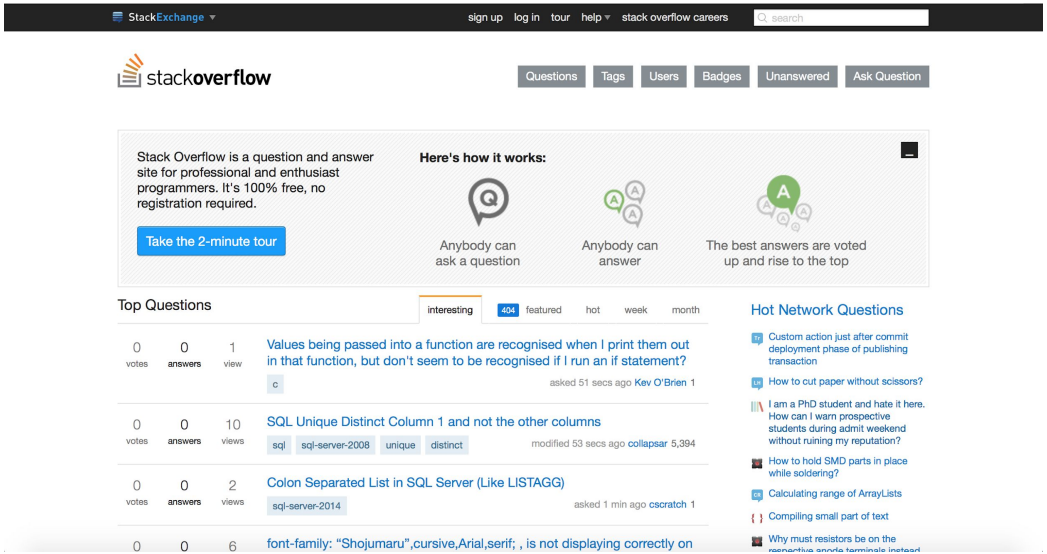
Manny C.

Stackoverflow

Millions of questions that need to be answered

An average of 11,000 questions asked each day

~16 million questions have been asked already



Lots of experts that are willing to get their name out there

There are ~3 million users that have logged in this year

~22 million answers have been provided for the ~16 million questions

How does the rest of the world use this forum?

- People who don’t post a question, nor don’t post an answer
    - Keyword search until they find a similar question to what problem they’re facing

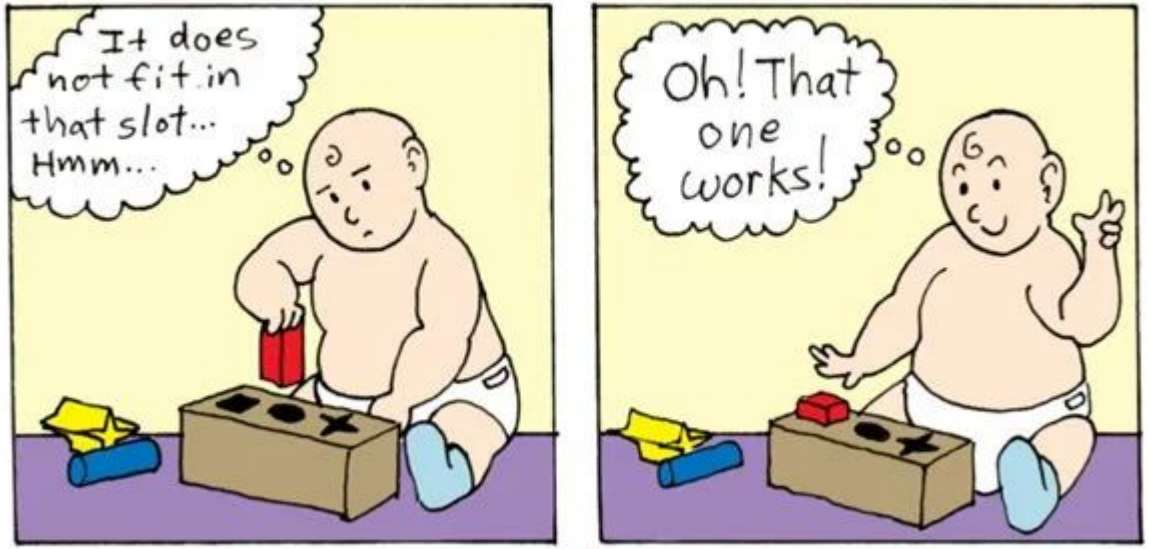
How do they find the right answer?

- They look for the accepted answer 
- They look for answers with upvotes 

- They can filter out bad answers if they have downvotes 

Other methods of filtering answers

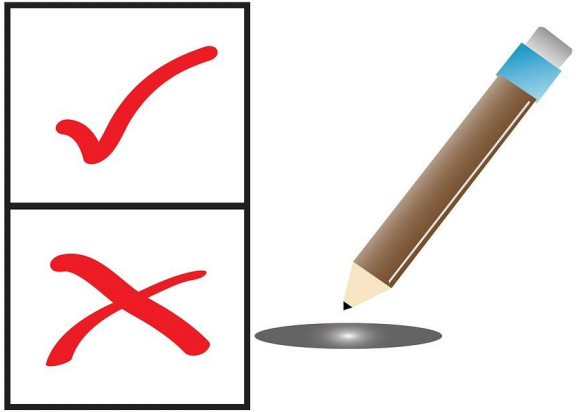
- Look through the comments of that answer
- Worst case: Trial and Error



Problem

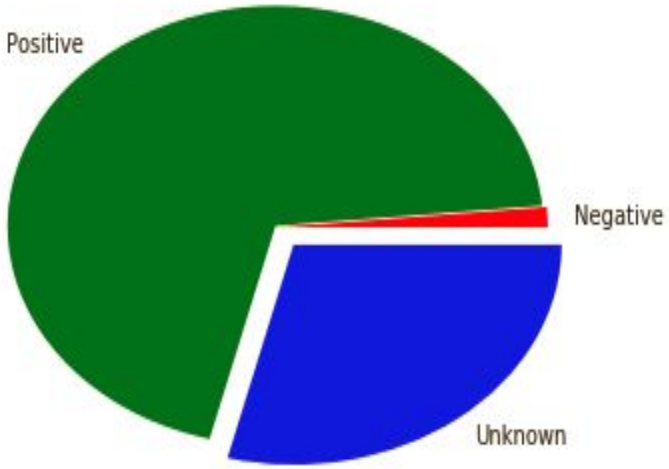
Proper answer relies on community input

- The original poster needs to mark the answer as accepted ‘voluntarily’
- Upvotes and downvotes are ‘voluntarily’ awarded to an answer
- Votes or Answer Acceptance are easy categories to filter out ‘right’ or ‘useful’ answers



Not everyone votes or Not every answer has votes

- Positive -> is Accepted answer or Has Upvotes
- Negative -> Has Downvotes
- Unknown -> No Votes Nor accepted



Answers

Proposed Solution

Improvement Potential

- Can we mark every question as being useful or not useful?
- In other words, if an answer has no votes, what can we say about it?

- This can improve the user experience for those that are looking for an answer

Solution Details

- If an answer has not votes: assign a yes or no based on the answer and user who answered it
- e.g.->

Based on the user and answer, StackoverFlow AI believes this answer is useful

0

I think that the problem you have here has to deal with the multiple return statements you have in that if statement block you mentioned.

This block is what you have...

```
if (!inputs.job.isAnalysisComplete) {
    return;
    actions.job.chargeToPersonalAccount();
    return;

    if (inputs.job.totalPages < 10) {
        actions.job.chargeToSharedAccount(ADM-3900);
    }
}
```

I think this block would be more accurate if it was something like this...

```
/*No details of print analysis?  Return the the function immediately!*/
if (!inputs.job.isAnalysisComplete) {
    return;
}

/*Job less than ten pages?  Charge shared account.  Otherwise charge personal account.*/
if (inputs.job.totalPages < 10) {

    /*Also, my bet is that the ADM-3900 needs to be in quotes for a string unless other wise

    actions.job.chargeToSharedAccount("ADM-3900");
} else {
    actions.job.chargeToPersonalAccount();
}
```

Note, this is my best guess seeing as I am not familiar with Papercut software.

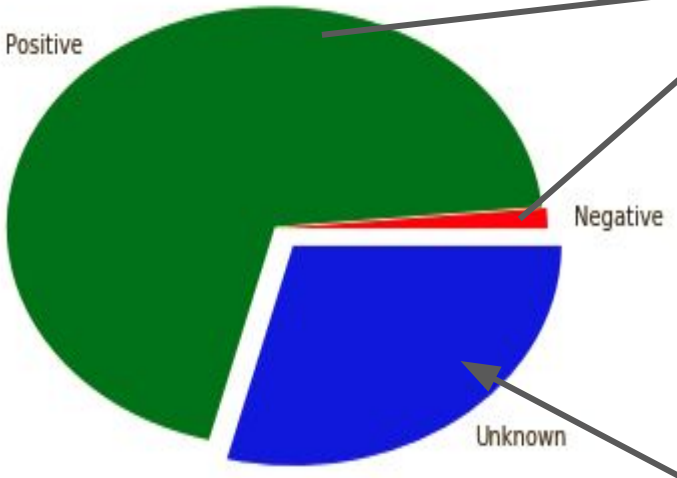
If that didn't help there is always technical support

Solution Details

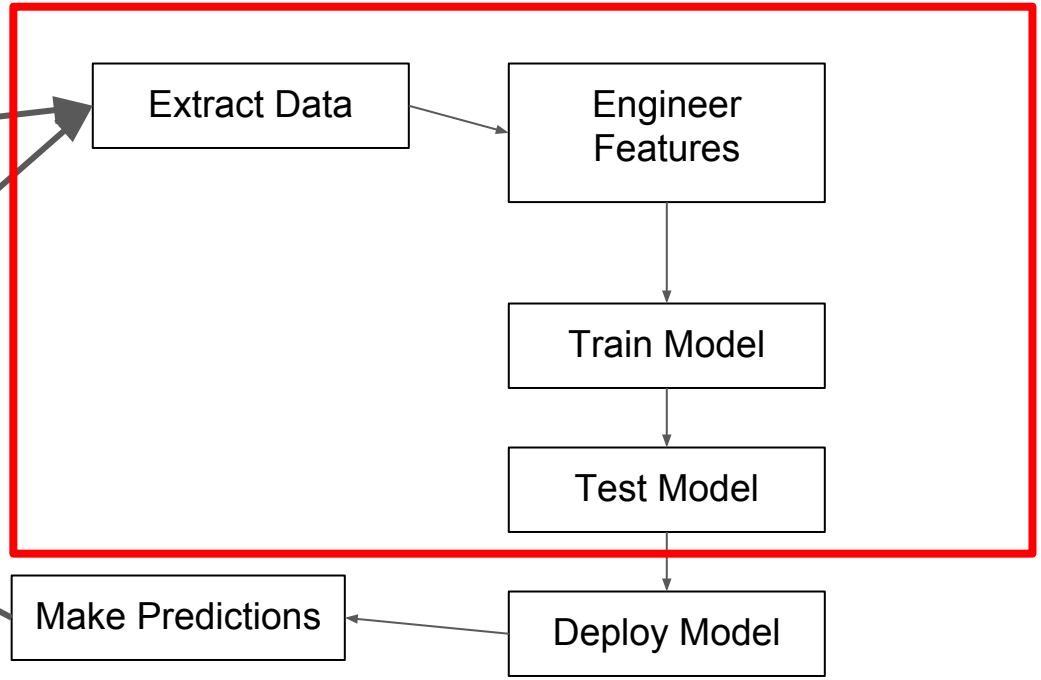
Let’s train this AI model

- Data extraction
- Feature Engineering
- Model Training
- Model Evaluation

Big Picture



Answers

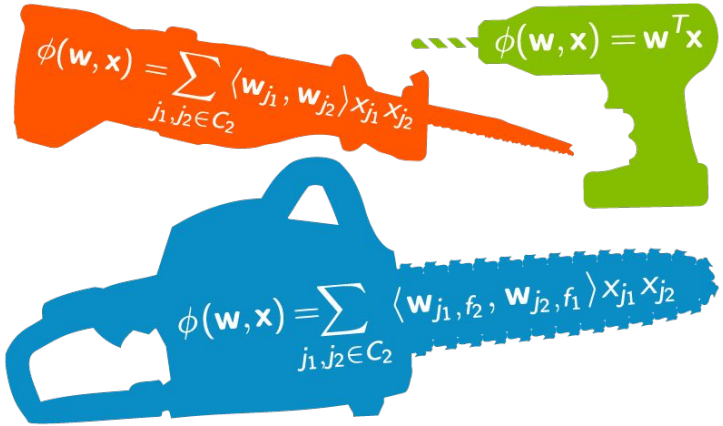


Data Extraction

- Extract all the data on:
  - Question Title, Question Body, Tags
  - Answer Body, comments, score
  - User History
    - All of it
    - Reputation
    - Badges
    - Comments
    - Previous answers
- Constraints:
  - Answer has votes or has been accepted

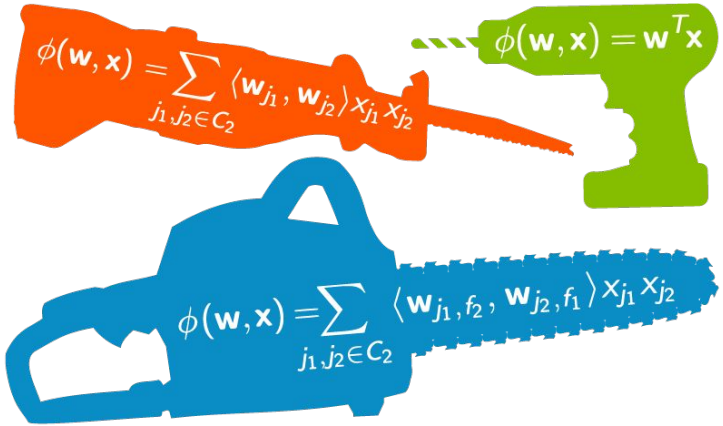
Feature Engineering

- Number of matched keywords between question and answer
  - Get rid of stop words (the, is, what, where, etc.)
  - Find number of matched keywords
  - NLTK
  - Also find total keywords



Feature Engineering

- Frequency of tags that match the users history and the question
  - Essentially trying to find how many times this user has answered questions with that tag.
  - Similar to above methodology
  - Also find total tags
- For user profile:
  - number of comments made
  - Reputation score
  - Number of badges
  - how many answers had code in them



Model Training/Evaluation

- Small portion of data was taken to keep costs down
- Standard Logistic Regression with L1 regularization
  - Binary classification
- Cross Validation to evaluate model performance

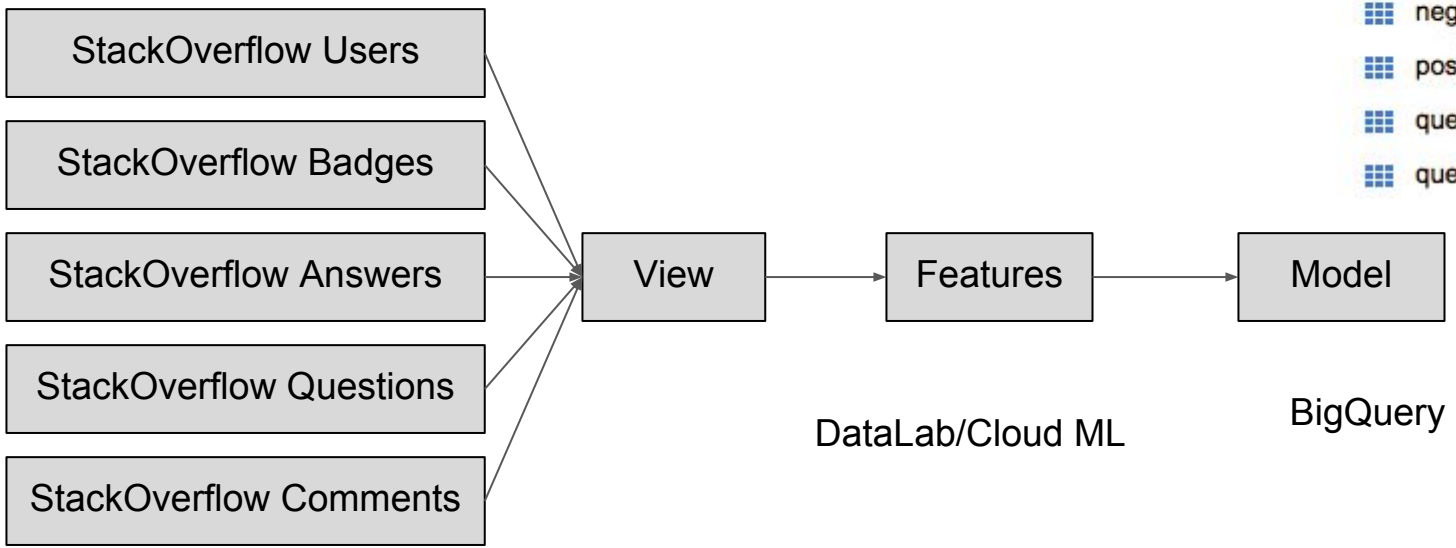
precision	recall	accuracy	f1_score	log_loss	roc_auc
0.61	0.59	0.60	0.60	0.68	0.64

What does this all mean?

- It’s really hard to predict if an answer will be useful
- To make it easier for ‘guest’ users to find the right answer, we may need a different strategy
- We can try a different model or engineer some better features to go further into an automated approach.

End of Slideshow

Data Pipeline



Google BigQuery

Model can be queried through BigQuery ML. It was the cheapest deployment option