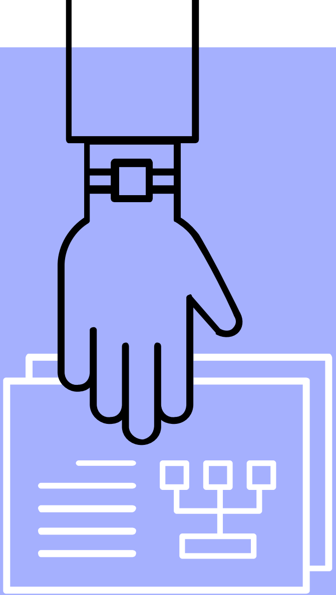
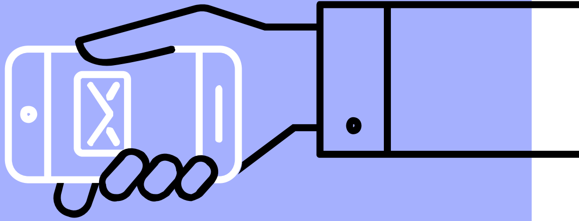
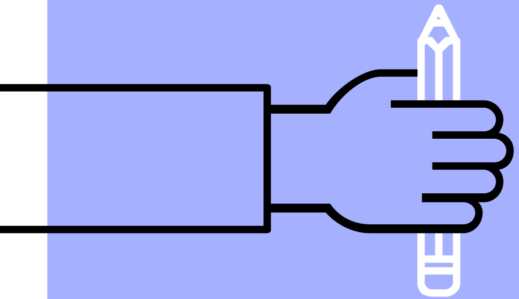
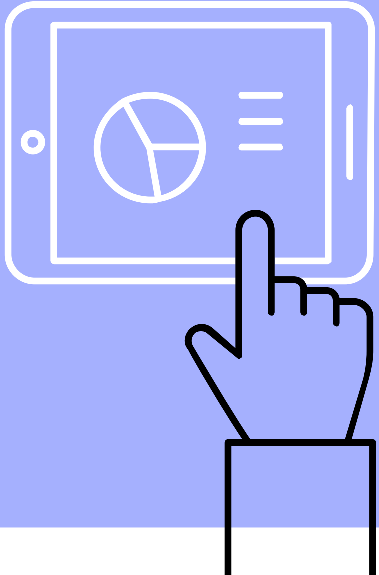




Question Pair Similarity





Describing the problem

Question Pair Similarity is a problem of finding pair of question that share the same semantics meaning.

Eg:

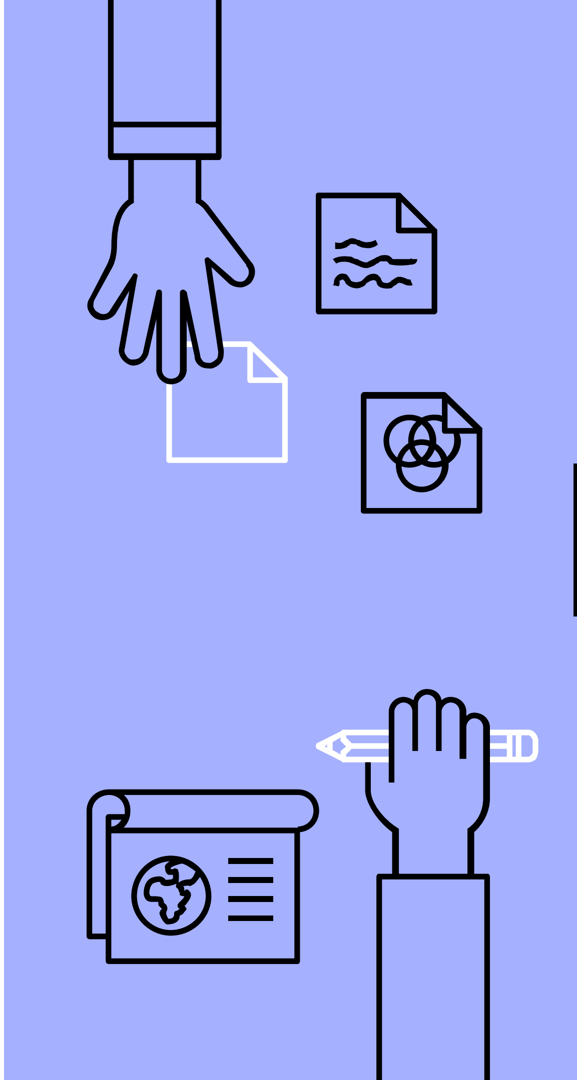
q1: is science and technology a blessing or a curse?

q2: is technology a blessing or a curse?

Application of Question Pair Similarity:

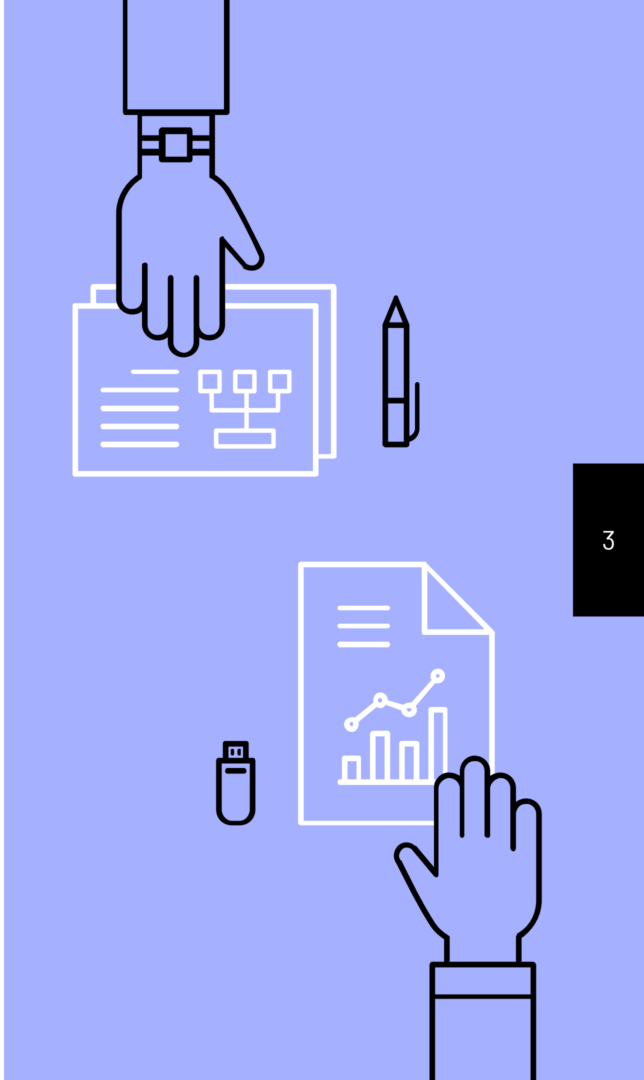
-Filtering duplicate questions in a question based website such as Quora, Stack Overflow or even Google

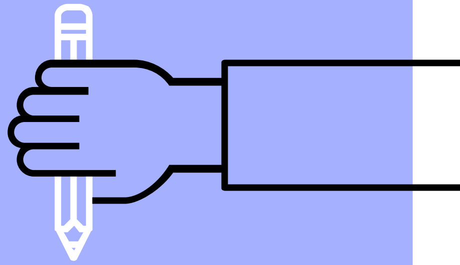
-Helps us to understand sentence semantic for natural language processing.



How can we solve this problem?

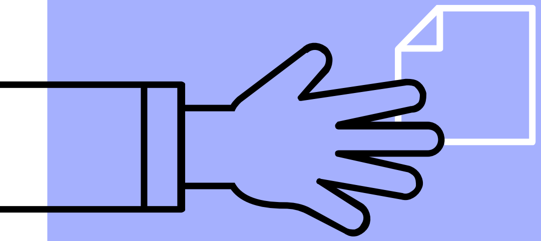
- Vectorizing words and sentences
- Select an appropriate classifier
- Pick appropriate independent variable





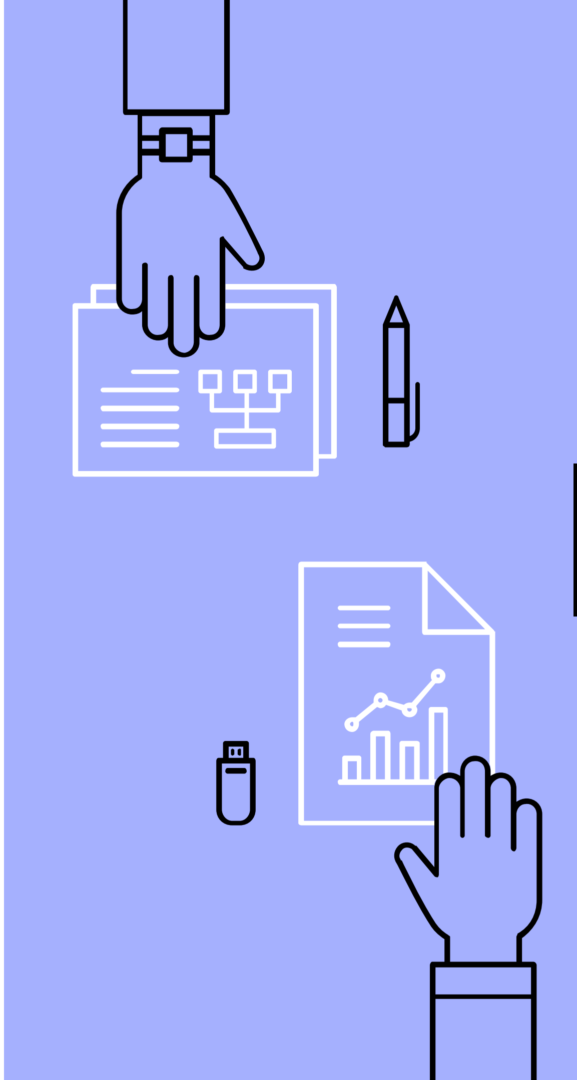
1.
Vectorizing Words and sentence

NLP->[0,1,0,0,1,1]



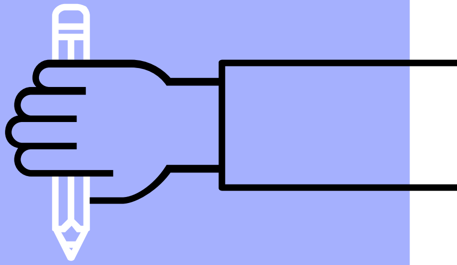
Tf-idf vectoring

- Each word in weight by how frequent it appears in the text divided by how frequent it appears in the documents
- Tf-idf algorithm creates a very sparse vectors (lots of zeros)
- Tf-idf algorithm creates a very high dimensionality vectors (size of vocab)



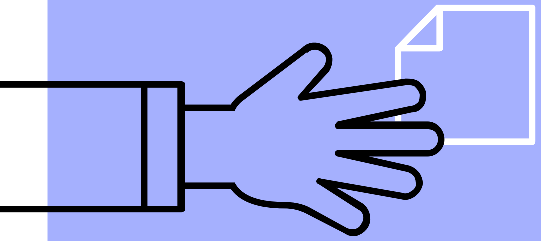
Word2Vec

- Word2Vec is a trained skip-gram model
- Word2Vec creates dense vector matrix
- Word2Vec creates vectors with lower dimensionality



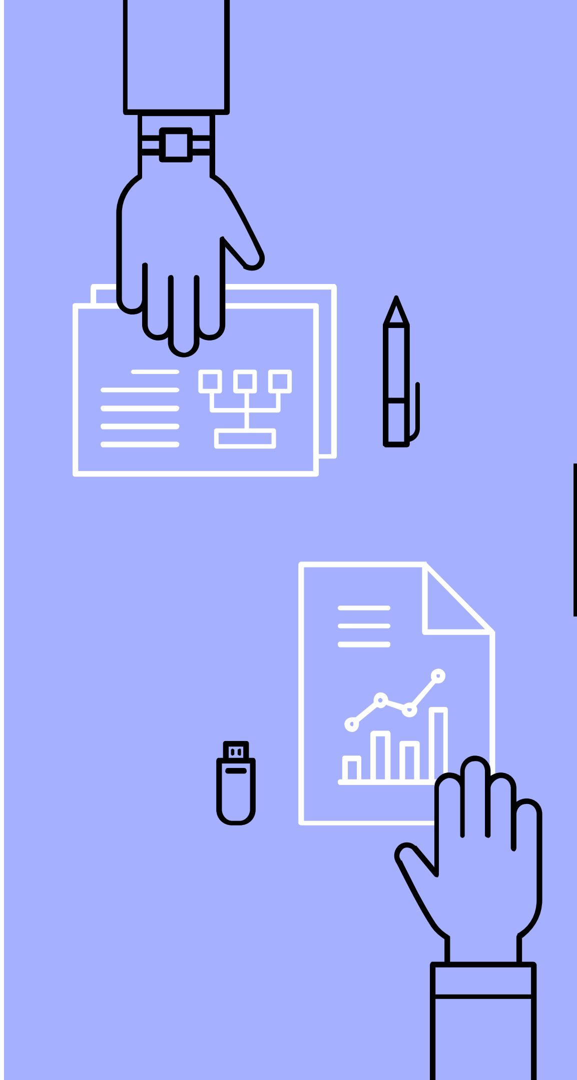
2.
Choose your classifier

Mirror mirror on the wall,
which is the best classifier
of them all



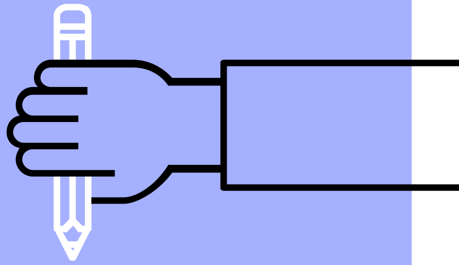
Naive Bayes

- Using log likelihood trained function to compute the likelihood that two questions are in the same class
- Each document in NB is the concatenate of the two question
- Naive Bayes is not really good for semantic purpose

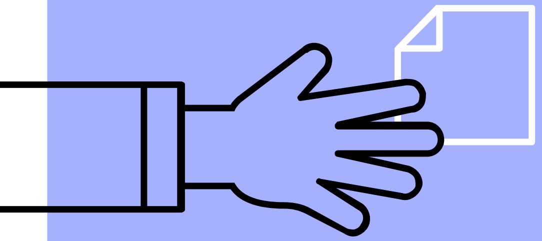


Logistics Regression

- Use multivariable Logistic regression on the different of the question vectors
- independent variable is the differences between q1_vector and q2_vector
- Using word2Vec is better than using Tf-idf algorithm because word2Vec have less zeros entry

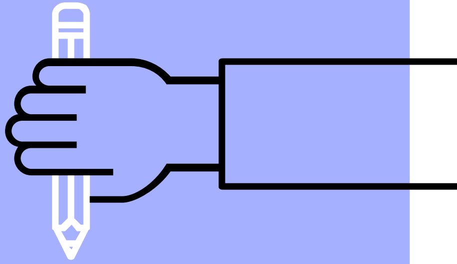


3.
Evaluation

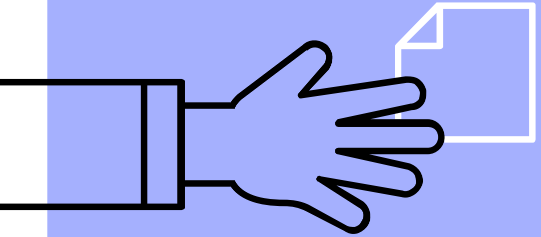


Comparison between models

	Naive Bayes	Logistics Regression	? classifier
Accuracy	70%	67%	~65%
Precision	10%	23%	~50%
Recall	71%	51%	~50%

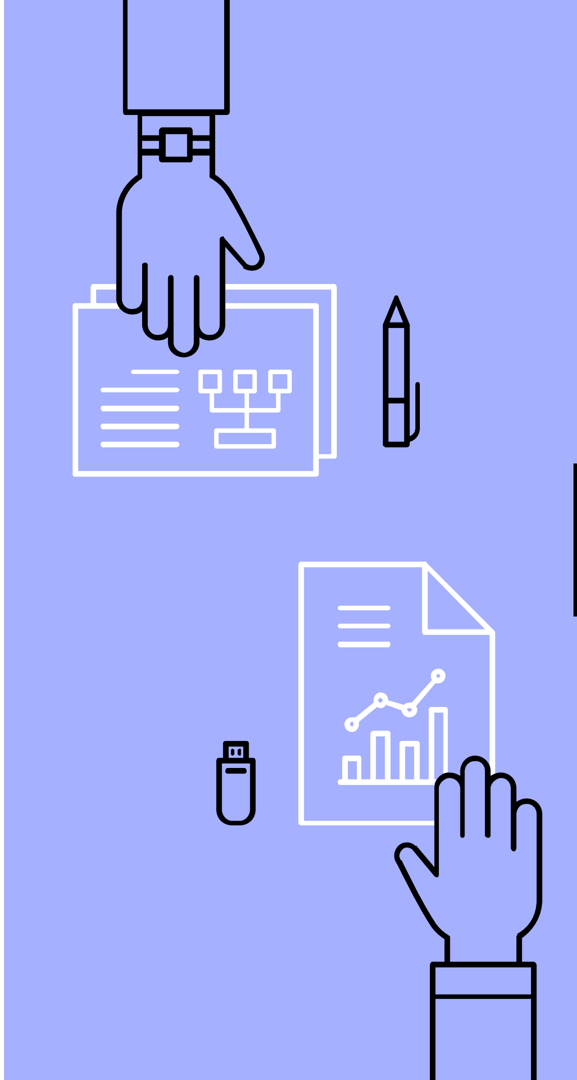


3.
How to improve?



More Data!

- start with 5000 questions
- Add up to 400000 and 3 more minutes of my time.
- Result: Every stats stay the same. However, we have a more consistent in accuracy, recall and precision

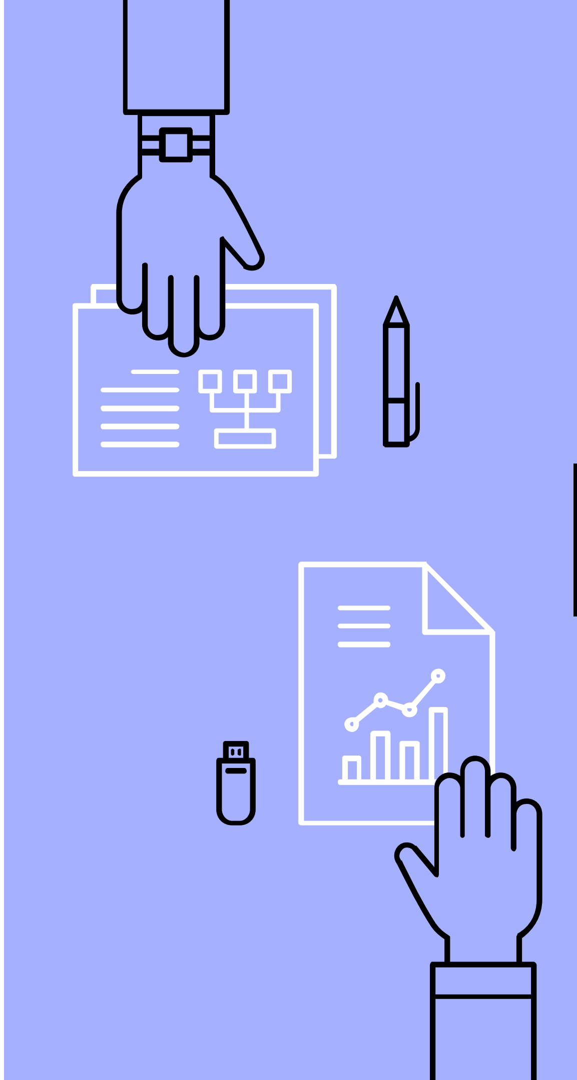


Brand new features!

- word match: how many word appear in both question1 and question2
- weighted tfidf: the weighted tfidf of the shared word between the two questions
- dot products: the dot product of two question vector
- Result: Success !! Accuracy : 73%, Precision: 45%, Recall: 53%.

XGB One Last Attempt

- XGB stands for eXtreme Gradient Boosting - A very powerful ML library
- Focus on speed and model performance
- XGB adds new model to fix the residual errors of the old model during the training process.



THANKS!

Any questions?