Identifying Duplicate Questions

With the growing popularity of question-and-answer sites/communities (Reddit, Stack Overflow, Quora, etc.), similarly worded questions have become a very common occurrence. Weeding out duplicates can provide a better experience on these sites.

Seekers often spend a lot of time searching for answers. Identifying and displaying similar questions that have already been answered, **saves time** and **increases satisfaction**.

Writers feel that they need to answer multiple versions of the same question.
Enabling them to tag multiple similar questions to their answer, allows them to **make more of an impact**.

Analytics Solution: A classifier to predict whether two questions have the same intent

Question Similarity Engine

1. Raw Data

400K Question Pairs

Duplicate Label

2. Feature Engineering

Natural Language Processing

- Number of Shared Words
- Difference in Readability Scores
- TF-IDF Vectorization
- Cosine Similarity
- Named Entity Recognition
- BERT Embeddings

3. Machine Learning

Classification Models

- Logistic Regression
- Random Forests
- Gradient Boosting

Hyperparameter Tuning

Evaluation Criteria

- Accuracy
- Recall
- AUC for ROC Curve

4. Interactive App

Recommender

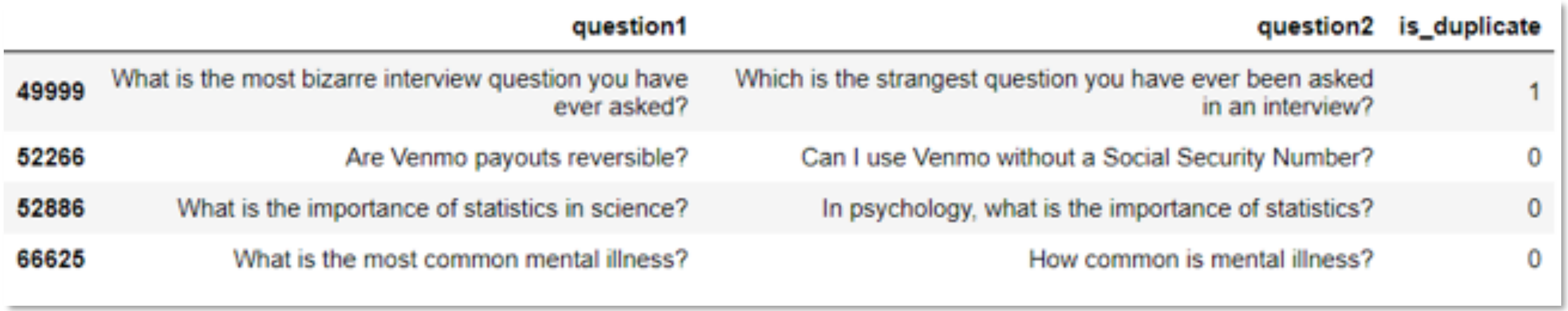
User Enters a question

- Model suggests top 5 similar questions based on predicted probability
- See python notebook: “Question Similarity Engine.ipynb”

1. Data: Quora Question Pairs

Since there are only two text input variables, feature engineering using NLP is critical

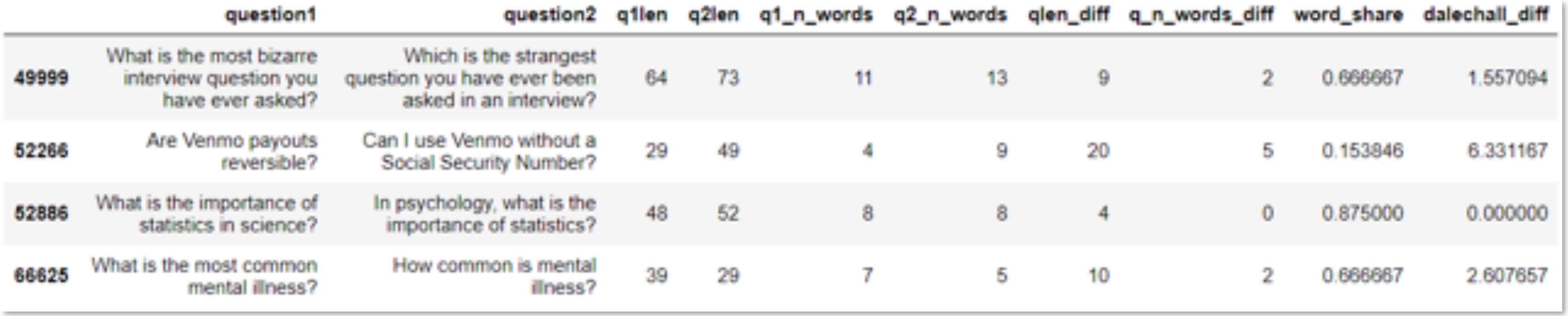
- Human labeled dataset with 400k pairs of actual questions asked on Quora
- 36.92% of this dataset contains duplicate pairs of questions



	question1	question2	is_duplicate
49999	What is the most bizarre interview question you have ever asked?	Which is the strangest question you have ever been asked in an interview?	1
52266	Are Venmo payouts reversible?	Can I use Venmo without a Social Security Number?	0
52886	What is the importance of statistics in science?	In psychology, what is the importance of statistics?	0
66625	What is the most common mental illness?	How common is mental illness?	0

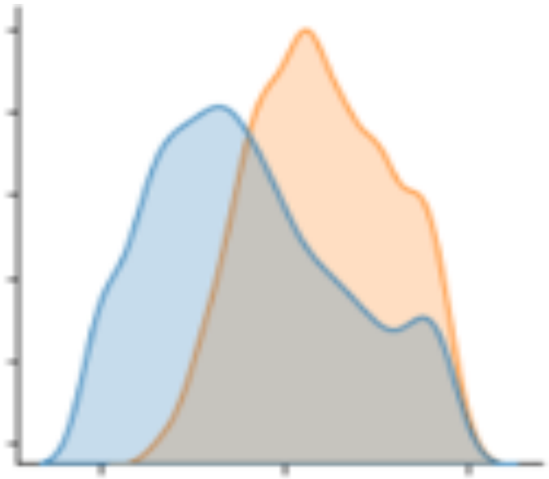
2A. Feature Engineering: Initial Variables

Initial variables were used to build a baseline logistic regression model, which yielded an accuracy of 67% an AUC of 0.73

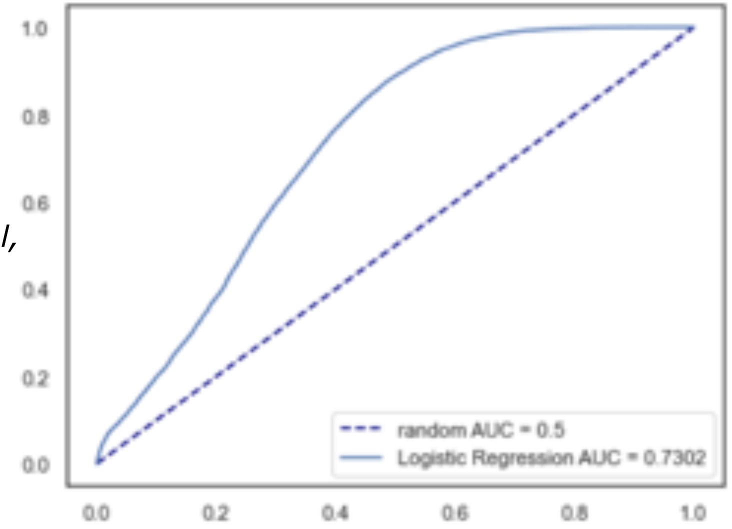


	question1	question2	q1len	q2len	q1_n_words	q2_n_words	qlen_diff	q_n_words_diff	word_share	dalechall_diff
49999	What is the most bizarre interview question you have ever asked?	Which is the strangest question you have ever been asked in an interview?	64	73	11	13	9	2	0.666667	1.557094
52266	Are Venmo payouts reversible?	Can I use Venmo without a Social Security Number?	29	49	4	9	20	5	0.153846	6.331167
52886	What is the importance of statistics in science?	In psychology, what is the importance of statistics?	48	52	8	8	4	0	0.875000	0.000000
66625	What is the most common mental illness?	How common is mental illness?	39	29	7	5	10	2	0.666667	2.607657

word_share vs is_duplicate flag

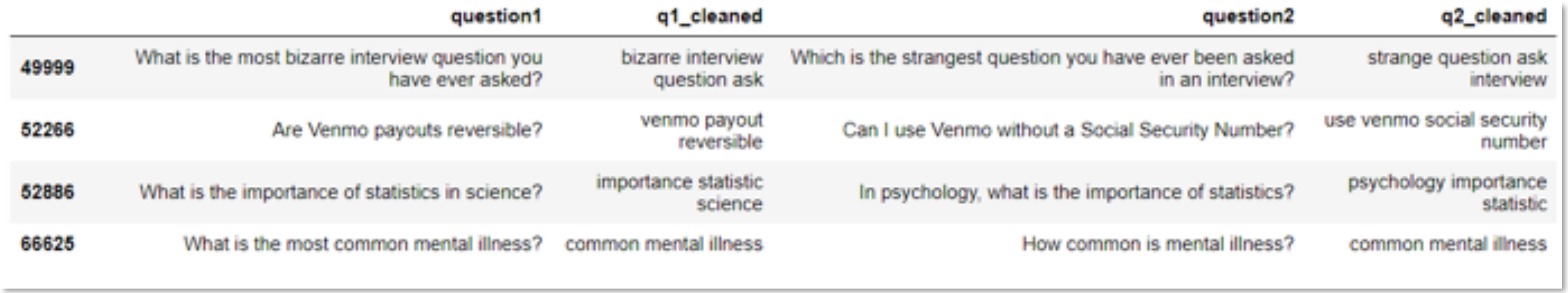


Initial logistic regression model, AUC: 0.73

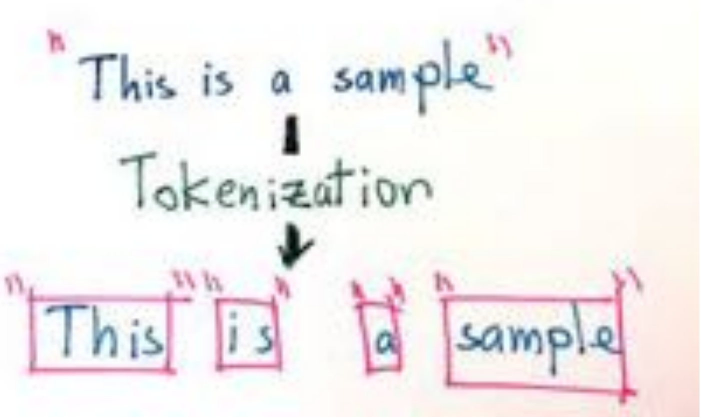


2B. Feature Engineering: Pre-processing

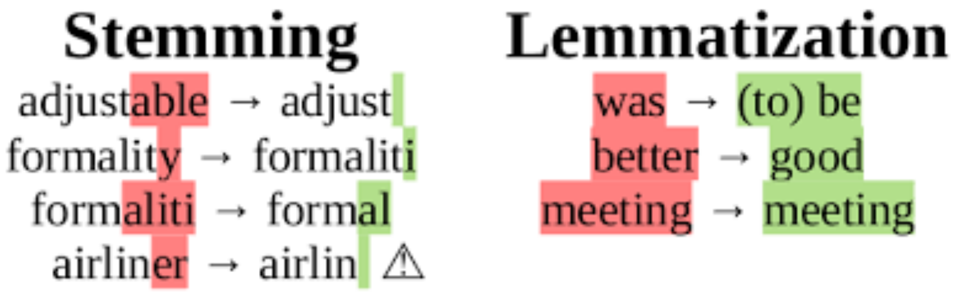
Questions were pre-processed by tokenization, lemmatization and removal of stop words using nltk and spacy libraries



	question1	q1_cleaned	question2	q2_cleaned
49999	What is the most bizarre interview question you have ever asked?	bizarre interview question ask	Which is the strangest question you have ever been asked in an interview?	strange question ask interview
52266	Are Venmo payouts reversible?	venmo payout reversible	Can I use Venmo without a Social Security Number?	use venmo social security number
52886	What is the importance of statistics in science?	importance statistic science	In psychology, what is the importance of statistics?	psychology importance statistic
66625	What is the most common mental illness?	common mental illness	How common is mental illness?	common mental illness



Tokenization



Stemming

adjustable → adjust
formality → formaliti
formaliti → formal
airliner → airlin ⚠

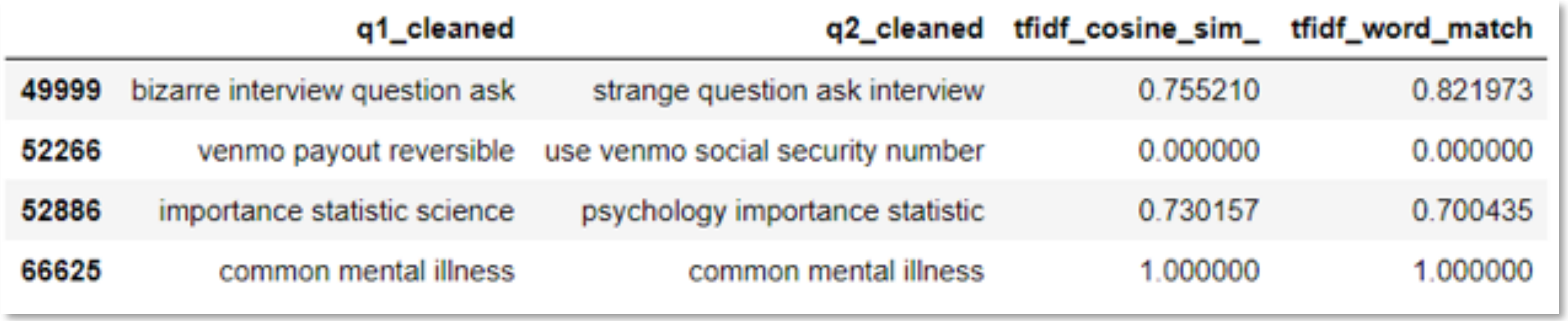
Lemmatization

was → (to) be
better → good
meeting → meeting

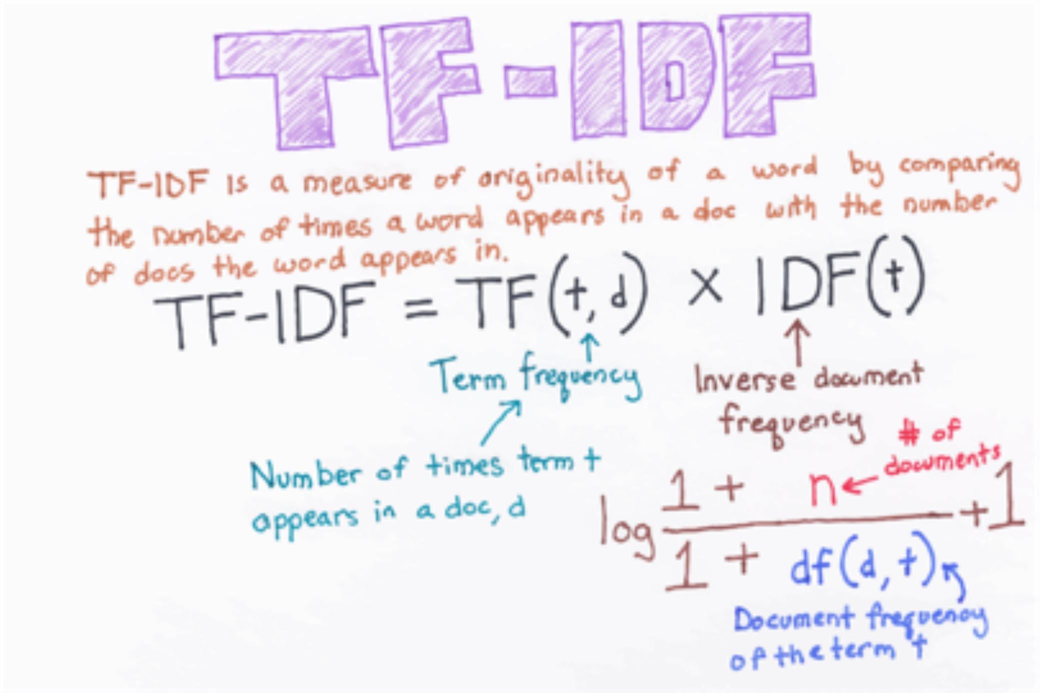
Stemming & Lemmatization

2C. Feature Engineering: TF-IDF Vectorization

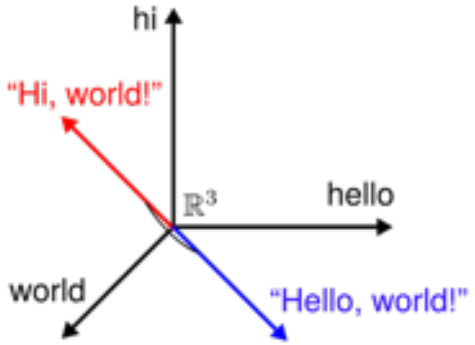
Tf-idf is a combination of term frequency and inverse document frequency. It assigns a weight to every word in the document, which is calculated using the frequency of that word in the document and frequency of the documents with that word in the entire corpus



	q1_cleaned	q2_cleaned	tfidf_cosine_sim_	tfidf_word_match
49999	bizarre interview question ask	strange question ask interview	0.755210	0.821973
52266	venmo payout reversible	use venmo social security number	0.000000	0.000000
52886	importance statistic science	psychology importance statistic	0.730157	0.700435
66625	common mental illness	common mental illness	1.000000	1.000000


TF-IDF
TF-IDF is a measure of originality of a word by comparing the number of times a word appears in a doc with the number of docs the word appears in.
TF-IDF = TF(t, d) × IDF(t)
Term frequency
Inverse document frequency
Number of times term t appears in a doc, d
# of documents
log (1 + n) / (1 + df(d, t)) + 1
Document frequency of the term t

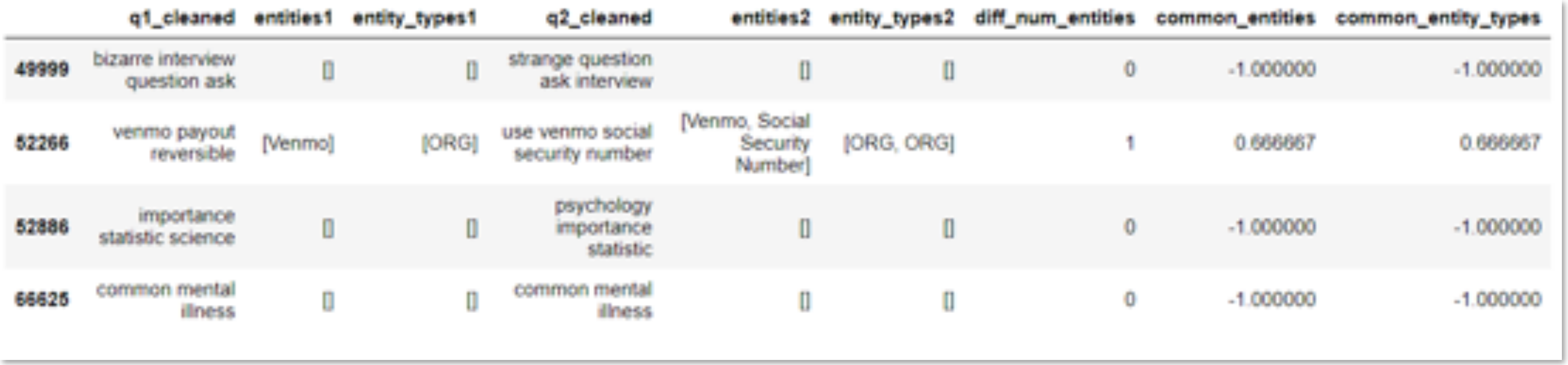
Calculating Similarity between two vectors



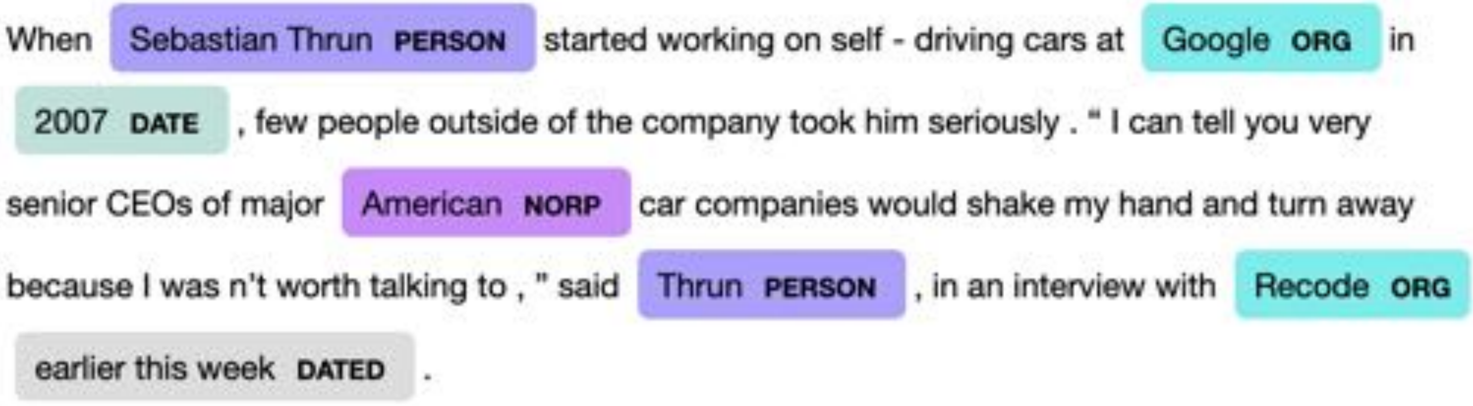
Cosine Similarity

2D. Feature Engineering: Named Entity Recognition

NER is an information extraction technique to identify and classify named entities in text, with the hope that questions with matching entities are more likely to be duplicates.



	q1_cleaned	entities1	entity_types1	q2_cleaned	entities2	entity_types2	diff_num_entities	common_entities	common_entity_types
49999	bizarre interview question ask	[]	[]	strange question ask interview	[]	[]	0	-1.000000	-1.000000
52266	venmo payout reversible	[Venmo]	[ORG]	use venmo social security number	[Venmo, Social Security Number]	[ORG, ORG]	1	0.666667	0.666667
52886	importance statistic science	[]	[]	psychology importance statistic	[]	[]	0	-1.000000	-1.000000
66625	common mental illness	[]	[]	common mental illness	[]	[]	0	-1.000000	-1.000000

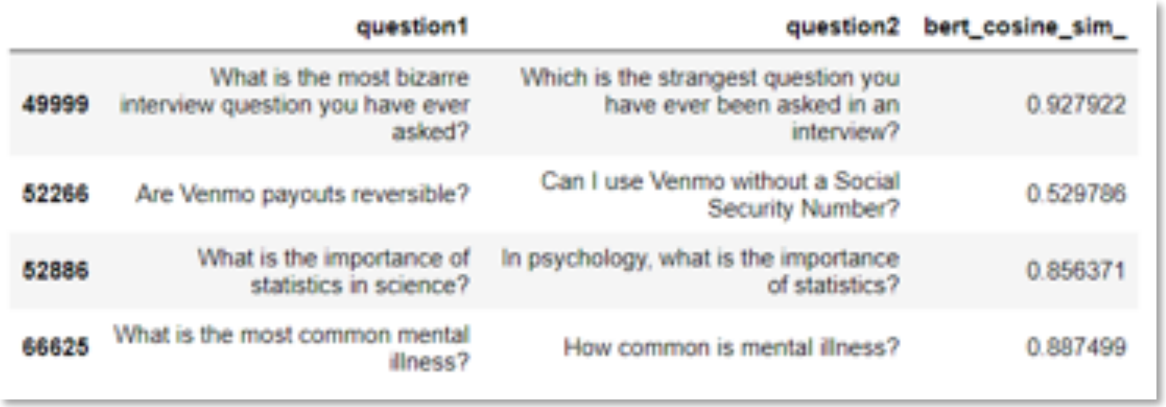


When Sebastian Thrun PERSON started working on self - driving cars at Google ORG in 2007 DATE , few people outside of the company took him seriously . “ I can tell you very senior CEOs of major American NORP car companies would shake my hand and turn away because I was n't worth talking to , ” said Thrun PERSON , in an interview with Recode ORG earlier this week DATED .

Color-coded recognized entities

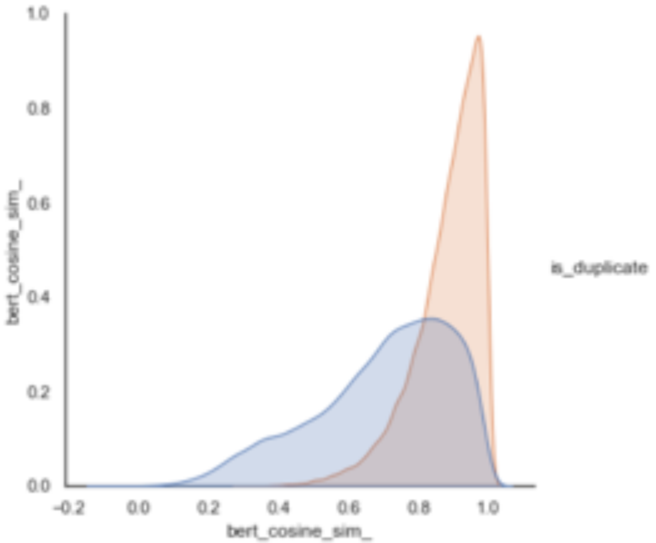
2E. Feature Engineering: BERT Embeddings & Cosine Similarity

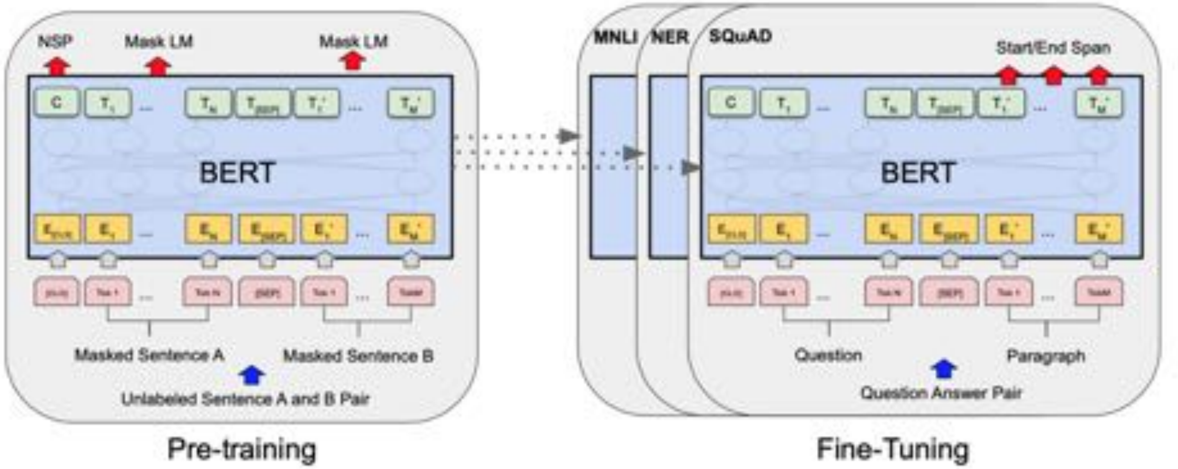
Huggingface’s pre-trained BERT base model, which has 12 layers (transformer blocks), 12 attention heads, 110 million parameters and hidden size of 768, was used to obtain embeddings



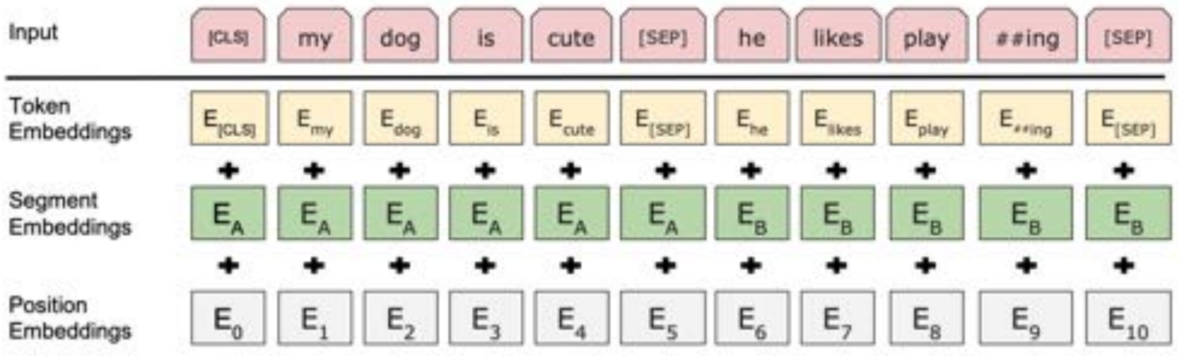
	question1	question2	bert_cosine_sim_
49999	What is the most bizarre interview question you have ever asked?	Which is the strangest question you have ever been asked in an interview?	0.927922
52266	Are Venmo payouts reversible?	Can I use Venmo without a Social Security Number?	0.529786
52886	What is the importance of statistics in science?	In psychology, what is the importance of statistics?	0.856371
66625	What is the most common mental illness?	How common is mental illness?	0.887499

BERT Cosine Similarity does a decent job of differentiating between class labels





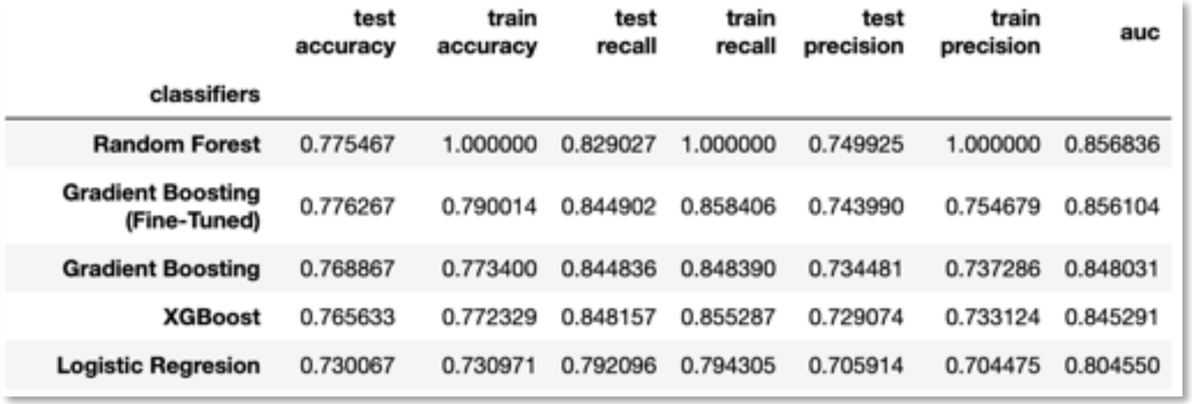
BERT training architecture (Image from https://arxiv.org/pdf/1810.04805.pdf)



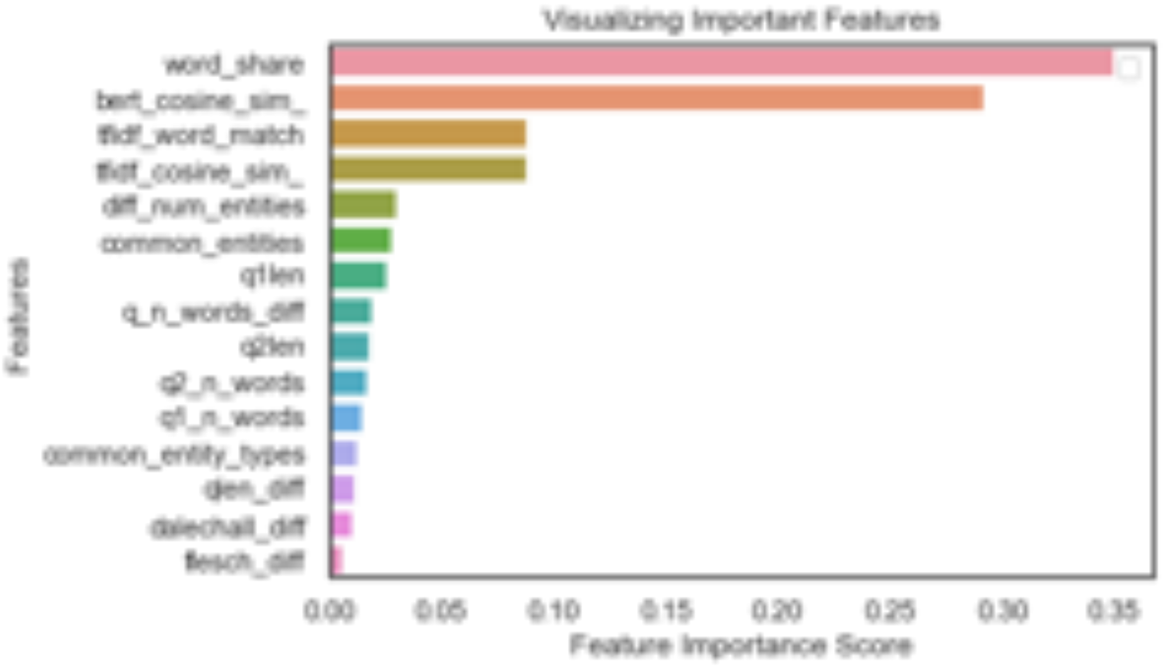
Input	[CLS]	my	dog	is	cute	[SEP]	he	likes	play	##ing	[SEP]
Token Embeddings	$E_{[CLS]}$	E_{my}	E_{dog}	E_{is}	E_{cute}	$E_{[SEP]}$	E_{he}	E_{likes}	E_{play}	$E_{\#\#ing}$	$E_{[SEP]}$
	+	+	+	+	+	+	+	+	+	+	+
Segment Embeddings	E_A	E_A	E_A	E_A	E_A	E_A	E_B	E_B	E_B	E_B	E_B
	+	+	+	+	+	+	+	+	+	+	+
Position Embeddings	E_0	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}

BERT input representation (Image from https://arxiv.org/pdf/1810.04805.pdf)

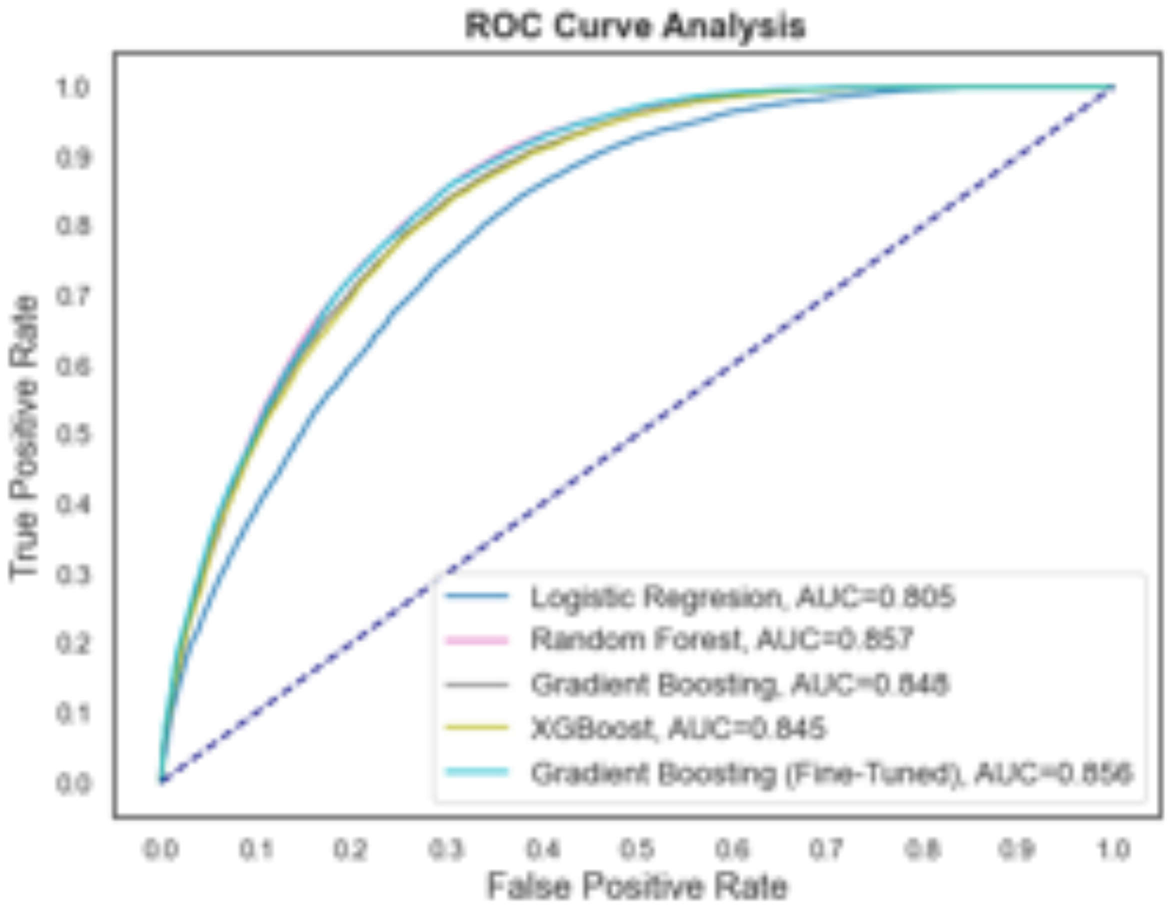
3. Machine Learning: Building & Comparing Classification Models



classifiers	test accuracy	train accuracy	test recall	train recall	test precision	train precision	auc
Random Forest	0.775467	1.000000	0.829027	1.000000	0.749925	1.000000	0.856836
Gradient Boosting (Fine-Tuned)	0.776267	0.790014	0.844902	0.858406	0.743990	0.754679	0.856104
Gradient Boosting	0.768867	0.773400	0.844836	0.848390	0.734481	0.737286	0.848031
XGBoost	0.765633	0.772329	0.848157	0.855287	0.729074	0.733124	0.845291
Logistic Regresion	0.730067	0.730971	0.792096	0.794305	0.705914	0.704475	0.804550



Visualizing Important Features

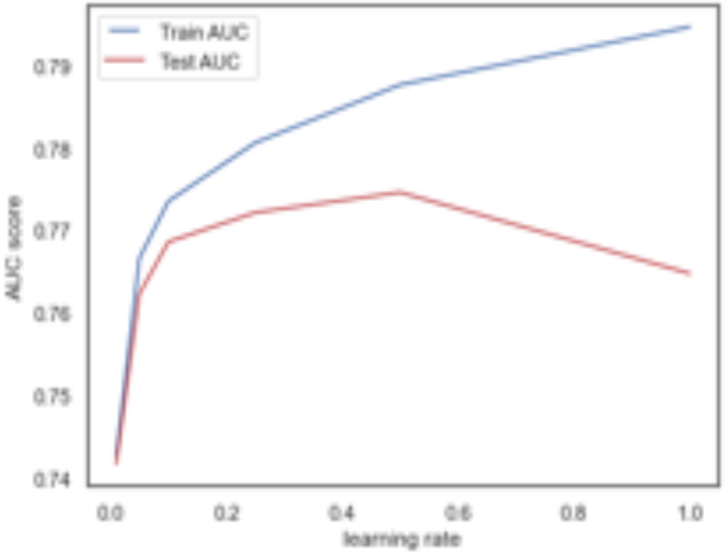


ROC Curve Analysis

3. Machine Learning: Tuning Parameters

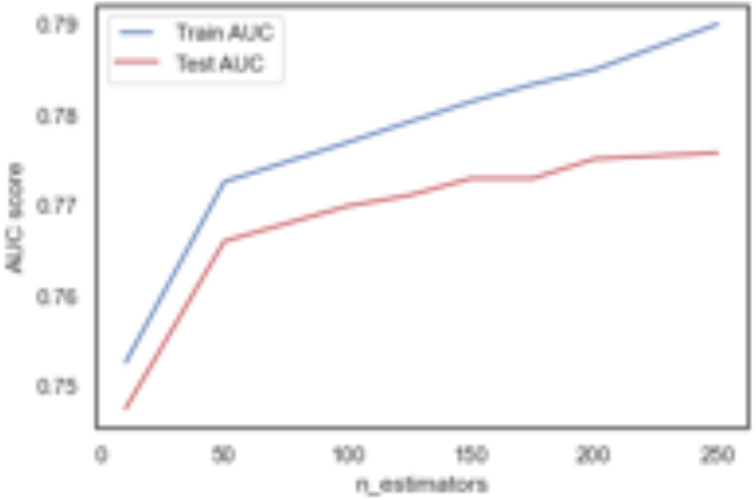
learning_rate

- This determines the impact of each tree on the final model
- Lower values make the model robust to the specific characteristics of tree but require a larger number of trees and are computationally expensive



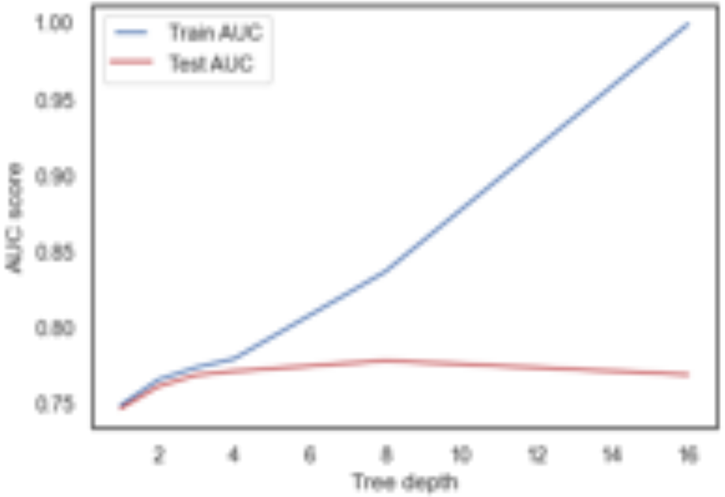
n_estimators

- The number of sequential trees to be modeled
- Though GBM is fairly robust at higher number of trees but it can still overfit at a point



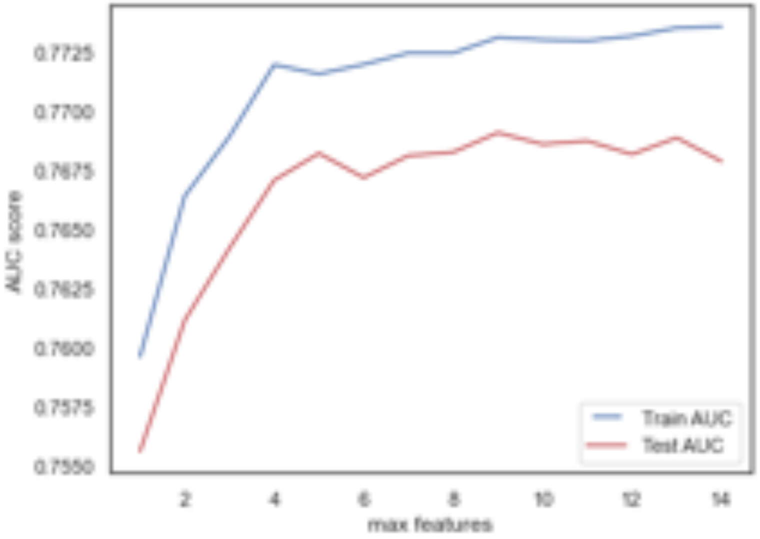
max_depth

- The maximum depth of a tree.
- Used to control over-fitting as higher depth will allow model to learn relations very specific to a particular sample.



max_features

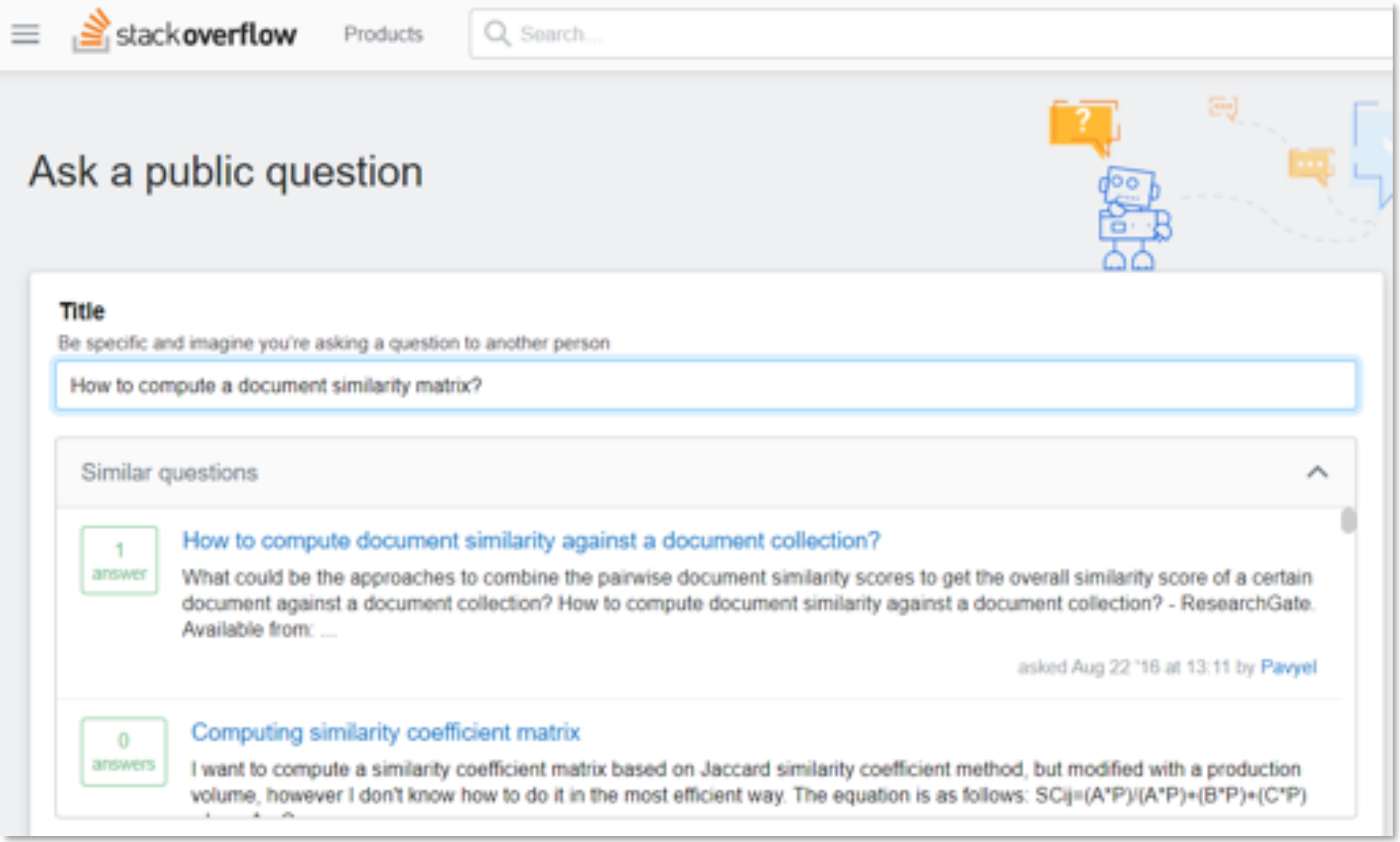
- The number of features to consider while searching for a best split
- As a thumb-rule, square root of the total number of features works great but we should check up to 30-40%
- Higher values can lead to over-fitting but depends on case to case



Final Model:
GradientBoostingClassifier(learning_rate=0.1,n_estimators=256,max_depth=4,max_features=5)

Applications: Q&A Websites

Q&A Websites like Quora, StackOverflow



Ask a public question

Title

Be specific and imagine you're asking a question to another person

How to compute a document similarity matrix?

Similar questions

1 answer

How to compute document similarity against a document collection?

What could be the approaches to combine the pairwise document similarity scores to get the overall similarity score of a certain document against a document collection? How to compute document similarity against a document collection? - ResearchGate. Available from: ...

asked Aug 22 '16 at 13:11 by Pavyel

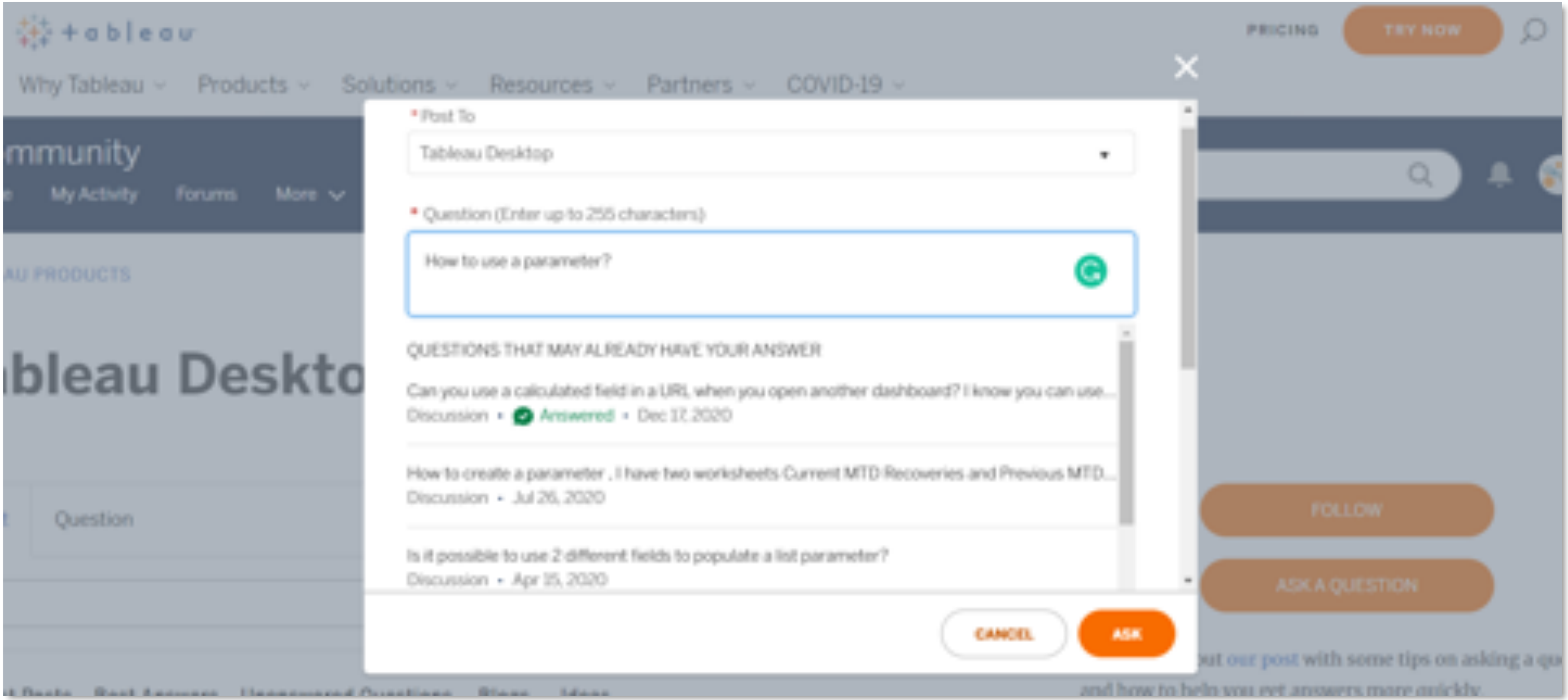
0 answers

Computing similarity coefficient matrix

I want to compute a similarity coefficient matrix based on Jaccard similarity coefficient method, but modified with a production volume, however I don't know how to do it in the most efficient way. The equation is as follows: SCij=(A*P)/(A*P)+(B*P)+(C*P)

Applications: Product Support Communities

Community sections for software/platform companies (such as Tableau) can leverage this solution to maintain a single source of truth and increase customer satisfaction



* Post To

Tableau Desktop

* Question (Enter up to 255 characters)

How to use a parameter?

QUESTIONS THAT MAY ALREADY HAVE YOUR ANSWER

Can you use a calculated field in a URL when you open another dashboard? I know you can use...
Discussion • ✓ Answered • Dec 17, 2020

How to create a parameter , I have two worksheets Current MTD Recoveries and Previous MTD...
Discussion • Jul 26, 2020

Is it possible to use 2 different fields to populate a list parameter?
Discussion • Apr 15, 2020

ut our post with some tips on asking a qu
and how to help you get answers more quickly.